University of Zurich[UZH]

Using Lexical-semantic Concepts for Fine-grained Classification in the Embedding Space

Thesis
presented to the Faculty of Arts and Social Sciences
of the University of Zurich
for the degree of Doctor of Philosophy

by

Michael Amsler

Accepted in the spring semester 2020
on the recommendation of the doctoral committee composed of
Prof. Dr. Martin Volk
PD Dr. Gerold Schneider
Prof. Dr. Wouter van Atteveldt

Zurich, 2020

"The only true wisdom is in knowing you know nothing."

Socrates

Abstract

The main contributions of this thesis are the following:

1. This thesis aims at bridging the gap between computational linguistics and the social sciences through the application of techniques from the former to research methodology of the latter. More precisely, we introduce an approach which links to established methodology in content analysis of textual material in social sciences but profits from the development in the field of distributional semantics. In addition to a concrete implementation, this work also deals with the following fundamental considerations:
   - The elaboration of common *pitfalls and specific challenges* in this interdisciplinary field forms the base for deriving general guidelines for designing such an approach.
   - The development of the approach is also influenced by more pragmatic decisions. We strive to foster the inclusion of as much *a priori-knowledge* as possible, especially to counteract the restrictions concerning the availability of annotated data. At the same time, the re-usability of the created resources are crucial design requirements.
   - The approach is nevertheless also suitable for *incorporating as much annotated data as possible* to learn from it in a data-driven fashion.
2. We introduce a new method for lexicon induction.
   - The proposed approach is suitable for *induction, improvement, and adaptation of lexical-semantic resources*, independent of specific domains.
   - The method is designed to be *versatile* for satisfying different needs of specific scenarios.
3. We introduce an application for classification based on these resources.
   - While producing comparable or superior results for the scenario of skewed data distribution (between different classes and with respect to the residual class), the approach follows the aforementioned crucial design constraints which are derived to satisfy the desiderata from the perspective of social science methodology. More precisely, especially *transparency, comprehensibility, modularity, and sustainability* are taken into account.
   - The classification is based on the sentence-level and offers fine-grained adaptation of the method and inspection of the results. This leads to an *understanding by analyzing* as a side effect.
   - The classification approach is basically *unsupervised*. However, automated tuning is applicable if additional annotated data is available.
   - The usage of *embeddings* (models of distributional semantics) is modular and thus offers an axis to tackle the challenge of application in different domains and for different languages. This leverages also resources and recent progress made in this field of research.

We make all our resources available at:

`https://pub.cl.uzh.ch/purl/ABCD`

Acknowledgements

When you embark on the adventure of a PhD thesis, you are usually not aware of where the journey can take you. If you find yourself in such a situation, it is helpful to have experienced navigators and helmsmen on board. I was able to rely on them during the research project NCCR democracy: Dr. Bruno Wüest and my supervisor Dr. Gerold Schneider "hired" me for this project and steered the boat through all wild waters and raging storms. I owe them my profound gratitude for making this journey possible.

Furthermore, we must not forget that research projects are not carried out in a vacuum but must always be supported institutionally. During my time as a PhD student and also before, I was allowed to be part of a team of researchers at the Institute for Computational Linguistics at the University of Zurich, which is characterized by dedication, competence, knowledge and variety. I am very grateful for this experience and can say without reservation that this home base has decisively enabled and shaped my voyage in research.

One of the most important cornerstones of research is the exchange with other researchers. This happens at conferences but also in daily conversations with other members of the institute. Here I would like to thank specifically Simon Clematide, Manfred Klenner, my supervisor Gerold Schneider and also our "Chief in Charge", Head of Institute Prof. Dr. Martin Volk—as well as all the other folks from our institute—for the support and advice they gave me in all these years.

At the same time, I would like to emphasize that for my interdisciplinary research the exchange with other researchers from other disciplines has always been very important and fruitful. This is especially true for all the collaborations with members of the Institute of Political Science and of the Institute for Communication Studies .

Of course, the personal environment as a source of motivation is just as important, and therefore I would like to thank my family for its moral support during all this time. This is especially true for Rebekka and little Malina, to whom I would like to dedicate this work.

I acknowledge the help of various proofreaders. I am thankful for all suggestions and comments. All errors remain my own responsibility.

Finally, I thank Wouter van Atteveldt who agreed to review this thesis and who accepted my invitation into the PhD committee.

Explanatory Notes: We Quote the Model

When writing a thesis, there are also some decisions to be made for which there is no clear-cut rule system to evaluate them as right or wrong. More precisely, these decisions are based on a grounding made of a continuum between opinion and reasonable argumentation. Hence, I will shortly try to explain my choice of expression throughout the thesis and illustrate briefly the somewhat special entry pages to the chapters.

When I started this thesis, I was confronted with the question of choosing the grammatical person to express myself. While it is nowadays also encouraged to use to first person singular (especially for single-authored scientific work), the more traditional way is to choose other means of expression—also, to draw attention away from the author and put emphasis on the distant and supposedly "neutral point of view".

But arguably, the strict usage of "*the author*" sounds awkward and is therefore not an option. Furthermore, if the passive voice is used (too) extensively or one switches thoroughly to the impersonal *one* the result does not feel very natural to me.

As an alternative, I chose to use the form of the first person plural for my thesis because I like the concept of the *inclusive we*.[1] This refers in my view to the idea that the writer and the reader(s) can form a *community* during the act of reading—which is an act of *commun*ication.

Although this act is temporally and spatially decoupled (and therefore the resulting community is decoupled as well), the community can set the scene for the picture of an author guiding the reader through its work like on a journey. Like taking someone on a tour and—being already thankful for the shown interest—henceforth using a friendly inclusive communicative form. This gut feeling is also enforced by the hunch that the readers of this work will rather be countable without using too many digits.

On the starting page of the chapters there is—next to the obligatory numbering and title—a quote, and, this is the special point I am referring to, a brief excerpt from an ipython-session[2]. This is an interactive mode of programming which comfortably allows for exhaustive experimenting, developing and creative application of resources to foster new ideas.

In these little pieces of ipython-sessions, normally a "query" (actually a function call or a statement) to an embedding model is shown, asking for an `analogy` or asking to display the `most_similar` instances, given the input. While especially the later option is

[1] See `https://oxfordediting.com/to-we-or-not-to-we-the-first-person-in-academic-writing/` for a more detailed discussion.

[2] See `https://ipython.org/`

one of the core properties used in this thesis over and over, the analogy case refers to the seminal paper by Mikolov et al. (2013b) describing `word2vec`, an algorithm to calculate such embeddings. In this paper, they show that the model also provides astonishingly appropriate answers to analogy questions, such as: a *man* is to *king* as *women* is to *X*. The model will then—based on simple vector addition and subtraction—return the most similar word in the semantic vector space for the query, in this case *queen*. This is especially worth mentioning because the model is trained only by processing huge amounts of raw text.

Since the mechanism works nicely for lots of examples, it may also be used to force the model to answer analogy questions for which humans would even have a hard time to figure out the relationships in between the given three input points. Sometimes the result will then be surprisingly poetic, just plausible, funny, or just rubbish (let aside the cases where the embedding reflects the biases of the corpus it was derived from).

Since this interface and this kind of representation and calculation has accompanied me on numerous days that I spent developing the approach presented in this work, I also like to show some examples as "quotes from the model", some intended to be surprisingly conclusive, others intended to be funny while simultaneously also pointing to the variability in quality.

However, I just would like to mention that the given examples are of course selected but not forged in any sense. They are the unmodified results of real coding examples.

Contents

List of Figures

List of Tables

List of Algorithms

1

Introduction

> "The covers of this book are too far apart."
> — Ambrose Bierce

```
In [163]: model.wv.most_similar(positive=["sozialwissenschaftlich",
"Computerlinguistik", "Idee", "Transfer", "Algorithmus", "Code",
"Brücke","bauen"], topn=5)
Out[163]:
[('Software', 0.5594435334205627),
 ('Computerprogramm', 0.5490928888320923),
 ('entwickeln', 0.530373215675354),
 ('Technologie', 0.5301876068115234),
 ('neuronalen_Netz', 0.528896152973175)]
```

1.1 Computational Linguistics Meets the Social Sciences

The application of methods from computational linguistics in social sciences is a flourishing research field. Since the performance of natural language processing has reached a promising degree of quality and the need for being able to process massive amounts of text has been strongly postulated for a long time (cf. Cheng et al. (2008)), the time is ripe to make concrete proposals to respond to these calls.

This thesis presents an approach which aims to avoid several pitfalls (cf. Grimmer and Stewart (2013), Shah et al. (2015), Van Atteveldt and Peng (2018)) in the application of language technology for the research process in social sciences which uses the methodology of content analysis.

It is one of the main goals to first derive a clear strategy to follow during this endeavour. More concretely, the development of methods and specific implementations attempts to follow guidelines which we establish in the first place. These guidelines should make sure that the proposed method respects several different aspects which reflect pivotal requirements of the target domain. The requirements in turn emerge on the one hand naturally from the specific task at hand, i.e., to solve a given problem: perform automated content analysis.

But the requirements comprise also of considerations which arise from the setting of interdisciplinary scientific work. More precisely, the endeavour to port methods and approaches for scientific workflows from one discipline to another always faces the challenge of acceptance in the target domain. It is up to the researcher who tries to bridge the gap between those disciplines to anticipate these possible difficulties and counteract them by means of designing and shaping the proposal in suitable ways.

Drawing on this premise, this thesis presents a novel approach especially designed to meet the requirements from social sciences with a focus on media and communication science as well as political science.

In the next section we will first outline the main research questions.

1.2 Research Questions

Starting with the derivation of central desiderata for approaches that are feasible to provide assistance to social scientists working with textual material, we strive to determine pivotal points for design decisions. Therefore, we ask,

- Which special requirements should be taken into account when proposing a technology transfer to another discipline? Which factors can contribute to a better acceptance and understanding in the target domain and which cliffs should be better circumnavigated?
- What are the central criteria for developing a solution for the transfer of techniques from computational linguistics to social sciences for the aim of a largely automated content analysis?

Chapter 2 will address these questions and serve as a base for further development. Building on these cornerstones, we will—after a short introduction of the foundations for the later applied methods—aim at answering the more concrete questions

- How can we facilitate the creation of lexical-semantic resources, especially for the case of a scientific study in which the phenomenon to be investigated is known a priori while the available data to investigate might only be sparse?
- Given the lexical-semantic resources created, how can we surpass the performance of existing approaches using such resources?
- How can we achieve interpretability for such an approach, considering transparency on the level of the model, its components, and its predictions?

While the proposed approach is described conceptually and algorithmically in detail in Chapter 4 and 5, it will be evaluated on different tasks in the empirical part of this thesis in Chapter 6, 7, and 8.

In a subsequent part (Chapter 9 and 10) we will discuss the advantages and possibilities of the proposed approach as well as its limitations and challenges. This part also links to newer developments in research which at least partly address some of the problems mentioned. Thus, we pose the following questions

- With a broader perspective: what are the main advantages of the proposed approach and which criteria of the initially created catalogue of desiderata can be satisfied in how?
- What are the limitations of the given approach?
- How can these limitations be mitigated?
- What additional benefits can be reached with this approach which are beyond the scope of this work?

1.3 Contributions

The main contributions of this work comprise the following:

- Collection and assessment of requirements:
  One of the core intentions of this thesis is the transfer of available technological

and methodological progress from computational linguistics to social sciences, especially for content analysis. Therefore, different requirements from the target domain are collected, discussed and subsequently taken into account to derive guidelines for the development of an apt implementation.

- A framework for
  - Creation and expansion of lexically-based concepts:
    In order to link to established methodology, we follow the general strategy to represent phenomena of interest with lexical resources. To foster the applicability of such resources, we introduce an instrument to derive and improve them in a semi-supervised way. More precisely, we use word embeddings (as a model of distributional semantics) to expand and enrich the lexical resource. This expansion is possible with or without involvement of the researcher conducting the content analysis, although the approach is especially designed for scenarios where "one knows what to find"—in the sense that conceptual and *a priori* knowledge may be integrated while still allowing for an optional integration of annotated data.
  - Adaptation of resources:
    In addition to the expansion, the approach also allows for quick adaptation and steering of the generation process. We will demonstrate this use case for a number of exemplary cases.
  - Application of the created lexically based concepts:
    While the generation of such resources can also be viewed as a task on its own, in this thesis, a lot of emphasis is put on the applicability of the resources for a wide range of tasks. To evaluate the approach, we establish two settings for which we present encompassing performance measurement and comparison to other approaches. Firstly, the "generic" problem of text classification (document-level) is put under scrutiny for settings with small annotated data sets which show a large distribution skew. Secondly, we turn to the more specific problem of framing analysis which is a widely applied approach to content analysis for media sciences as well as political sciences.
- An outlook on further application scenarios and a suggestions for coping with commonly encountered problems:
  Since re-usability and modularity are two core requirements that have to be met, we also propose a set of appropriate techniques (and implementations) to tackle challenges which can emerge through the dependency on models of distributional

semantics (embeddings), such as out-of-vocabulary problems and domain adaptation. Additionally, we also propose a solution for cross-lingual applicability of the resources.

1.4 Outline

The chapters of this thesis are structured as follows: first, we focus more thoroughly on the problems faced when porting methods from computational linguistics to the social sciences and elaborate a clear argumentation frame for deriving design principles (Chapter 2).

Second, we shed light on the various techniques and methods that are applied in the subsequent development of the pipeline, namely the models of distributional semantics and the emergence of the dominating vector-space word models (embeddings), lexicon induction, and classification with a focus on lexical-semantic resources (Chapter 3).

Third, we describe in detail an approach that in a first step simplifies the generation of lexical-semantic resources (Chapter 4) and in a second step allows their application to core content analysis methods like fine-grained classification on texts (Chapter 5).

Fourth, we evaluate the derived pipeline on different empirical scenarios, starting with pure lexicon induction (Chapter 6), followed by the application of lexical resources in classification scenarios with skewed data problems (Chapter 7), and lastly report on the results of the application for fine-grained accountability framing detection (Chapter 8).

The results are put under scrutiny to derive findings about their general applicability (Chapter 9). Furthermore, we devote a chapter to open problems and future work to alleviate shortcomings of the approach, in which we also highlight current research in the specific fields (Chapter 10).

Lastly, the approach, the empirical results from the experiments, and the findings are summarized in the conclusion (Chapter 11) to answer the research questions.

2

Crossing Fields: Problems and Challenges

> "There is only one way [...] to get anybody to do anything. And that is by making the other person want to do it."
> — Dale Carnegie

```
In [125]: analogy(a="Computerlinguistik", b="Algorithmus",
x="sozialwissenschaftlich", y=None, model_given=model, verbose=True)
'Computerlinguistik' is to 'Algorithmus' as 'sozialwissenschaftlich' is to 'methodisch'
Out[125]:
[('methodisch', 0.4618265628814697),
 ('empirisch', 0.45500725507736206),
 ('Fragestellung', 0.420206755399704),
 ('soziologisch', 0.4168052077293396),
 ('quantifizierend', 0.41447246074676514),
 ('mikroökonomisch', 0.41426777839660645),
 ('Big_Data', 0.4100503921508789),
 ('verhaltensökonomisch', 0.40668728947639465),
 ('heuristisch', 0.40540611743927),
 ('komplex_mathematisch', 0.3975958824157715)]
```

In this chapter, we will elaborate briefly the central problems which can emerge when applying specific methods from computational linguistics. We focus not only on the pitfalls which need to be avoided but also strongly on the further desiderata for such

approaches that lead to the applicability in multiple cases as well as their re-usability in comparative settings.

2.1 Computational Linguistics for Content Analysis in the Social Sciences

2.1.1 A Word on Interdisciplinary Research

An attempt to bring methods from one discipline to another is always a daring venture. Interdisciplinary research is about building bridges between the source and the target field[1]. This leads to the requirement that most importantly, the proposal must find acceptance in the target field—or it will simply be ignored. To match this criterion, we need to understand the basic demands and the methodological framework to which we want to contribute to.

When we look at this framework and identify its most crucial points—where a violation would render the proposal unacceptable in the target domain—there will also be requirements which we *could* take into account to make the attempt of bridging the gaps more promising. For example, it is an incentive for adoption in the target field if we solve a long lasting problem or remove an annoyance of the methodology.

Hence we strive to fulfill the identified most important criteria and to follow the goal to include such incentives as we have mentioned. Following this strategy, we almost certainly meet the point where those demands from the target discipline will in turn pose requirements for the method or technology from the source discipline.

This is where the second peril is lurking: if the adaptation of the method or approach leads to a form which is not of interest given the view from the source field (or not considered as a valuable contribution) the outcome will be that we have not built a bridge but just carried out a transport (of the method) and an imposed transformation. Or, more precisely and sticking to the metaphor, we would just jump over the gap and then have no connection back to the side where we came from. If our goal is not to build a single-point application of rather ephemeral nature, we need to take care that the contribution of this work is also visible for the source discipline.

This leads us to the following two main goals which form the pillars of this work. First, we want to maximize the probability of adoption of the method in the target field.

[1] We will refer here to source and target where we mean the field from which the methods that we port stem from and the field to which one wishes to introduce them. In our case, computational linguistics is therefore the source discipline and the social sciences—if they use the method of content analysis for textual material—are the target discipline.

Therefore, we apply a double-ended strategy. On the one hand, we need to comply with the most important criteria of the methodological framework of the target field. These points will be addressed in the sections 2.1.3 and 2.1.4. On the other hand, we incentivize the adoption of the approach through linking to established methodology while at the same time tackling persisting cumbersome problems involved in these methods. We elaborate on this in the next section.

Second, and rather naturally, in order to contribute to the research of the source domain (computational linguistics), we need to propose an approach consisting of novelty or improvement in a given comparison. Of course, the combination of both features is a desirable outcome. We therefore set these achievements as a second goal for this work.

2.1.2 A Link to Established Methodology: Dictionary-Based Content Analysis

As we have mentioned, we would like to incentivize the adoption in the target field by linking to established methodology. When we turn to computer-aided text analysis (CATA) as a method of content analysis, and, more specifically, to fully automated (often dictionary-based[2] approaches, Neuendorf (2016, p. 147) states that this is not a new procedure since it has been introduced more than half a century ago with the General Inquirer (Stone et al., 1966) and has been widely used since then.

In fact, the automated methods are so often applied that in her evaluation (Neuendorf, 2016, ibd., (emph. in orig)) even refers to such schemes of CATA as "a set of techniques that have become so common that it's rare to find a text content analysis today that does *not* use some type of CATA"[3].

One reason for this far reaching diffusion of the method is certainly that it is conceived as "perhaps the most intuitive and easy to apply automated method" (Grimmer and Stewart, 2013, p. 274). For instance, for document classification "it is based on a list of words or phrases and their associated category labels" (Petchler and González-Bailón, 2015, p. 435). Those lists are then consulted for a look-up for each document under

[2]Some other scholars refer to these methods as "lexicon-based", e.g. Petchler and González-Bailón (2015, p. 435f). Since the according literature from computational linguistics about the work in this field—prominently in sentiment analysis—often refers to those resources as lexicons, we will stick also to this term in our thesis (see also Chapter 6).

[3]See also (Neuendorf, 2016, p. 146f), (Grimmer and Stewart, 2013, p. 274), and (Petchler and González-Bailón, 2015, p. 435f) for a number of references to such analyses.

scrutiny and the rate of occurrence is hence used to measure into which category a document will fall or to which extent (Grimmer and Stewart, 2013, p. 274)[4].

As widespread as the usage of the method may be, researchers also warn against just using off-the-shelf dictionaries and emphasize the need to adopt them, or even create them from scratch for the task at hand (Grimmer and Stewart, 2013, p. 275). This process of adoption and curation, or even creation of such resources from scratch (or at least of careful validation in cases where pre-existing resources are applied) is generally accepted as necessary. However, this step of resource development is considered as "typically a long, iterative, and painstaking process" (Neuendorf, 2016, p. 149). Although there are tools which support the researcher in this work, "the process is still arduous" (Neuendorf, 2016, ibd.)[5].

In conclusion, we observe that this method is widely employed and accepted (although it is also considered simple), but it has inherent shortcomings in that the resources are not truly apt for the task at hand, or their adoption or creation is considered cumbersome. Thus, we identify this as an ideal point to link to this established method while at the same time proposing a solution to alleviate the problem of their curation.

2.1.3 Central Desiderata: Validity and Reliability

In order to comply with the most important criteria of the methodological framework in the target field, we have to account for validity and reliability.

Reliability is an important criterion as it assesses a measuring procedure on its extent to produce the same result for repeated trials (Neuendorf, 2016, p. 19). In other words, it describes how good our instrument (the content analysis) works in the sense that it produces the same output for the same input. In the case of manual content analysis,

[4] Interestingly, scholars like (Bholat et al., 2015, p. 1) refer to the form of reasoning and link such approaches to a deductive scheme of research, because "start with a predefined list of words, motivated by a general theory as to why these words matter" and sees them in counterposition to other (mostly unsupervised) methods which they deem as abductive.

[5] We would like to point out that we are well aware of the research direction of computational communication science (see Van Atteveldt and Peng (2018), Hilbert et al. (2019)) that has already embraced other techniques for automated text analysis which are more sophisticated and powerful. However, since dictionary-based approaches are simple *and* widespread, we use this method as an anchor point to posit our incentive in order to reach a larger audience. Nevertheless, we believe that the described approach holds also great potential for more complex architectures and technology, namely in a hybrid conjunction with transformer models such as BERT which was introduced by (Devlin et al., 2019) .

this considers the question of how well two (or more) coders[6] agree on the coding of a unit of analysis and which is referred to as intercoder reliability (Neuendorf, 2016, ibd.).

While intercoder reliability is typically measured between two coders and is hence directly linked to repeated analysis of the same unit, there are at least two ways in which one could integrate this criterion into automated content analysis. First, we could conceive the automated process as another coder (Zamith and Lewis (2015) uses the term "algorithmic coder"). We could then further compare this "coder" to other human coders to assess the level of agreement and report this as reliability. But in most of the work in automated content analysis, these elements of quality assessment are discussed under the aspect of validity (see below). Second, if we just apply the same criterion of reliability to the automated coder as we do for human coders, then the answer becomes simple. As long as the programmatic approach is deterministic, the output will be always the same for a repeated trial with a given input. Following such a conception, there is no need to assess the intercoder reliability of an automated approach further (cf. (Zamith and Lewis, 2015, p. 311)), as there will be full agreement on different runs[7].

Validity, on the other hand, is understood as "the extent to which a measuring procedure represents the intended—and only the intended—concept" (Neuendorf, 2016, p. 122). More concretely, in the case of automated content analysis, validity refers to the extent to which the output of our automated approach is in agreement with a given gold standard, which is typically a manually coded data set that contains supposedly "correct" codings (Zamith and Lewis, 2015, p. 311).

As a reader with a computational linguist's point of view will identify this as the frame of evaluation (of performance for a given method and data set), it is important to note that we subsume this under the criterion of validity[8]. In addition, the benchmark with competing approaches (or at least with a baseline)—which is a standard in the realm of empirical work in natural language processing—is claimed to have no counterpart in traditional content analysis (Zamith and Lewis, 2015, ibd.). Nevertheless, this becomes

[6] In social sciences, the act of annotation is called *coding*, and hence *coders* are the equivalent of *annotators* in computational linguistics. There is no overlap with the meaning of coding as writing programs that are executed by a computer. Since we are here referring to the methodological framework from social sciences, we consider it to be more precise to use the original terminology. However, in the later chapters, we will use the term "annotated data" or "labeled data" as we are then positioned in the realm of computational linguistics.

[7] Although this argumentation is perfectly fine, one might be cautious because the premise of deterministic behaviour is not always given. For example, a classifier which has to decide for the category of a text may be faced with two equal sized quantities (e.g., similarities or distances to centroids or decision boundaries). Typically the assignment is than arbitrarily chosen. As this situation may not occur often in high-dimensional vector spaces, more simple accounts for classification, relying on few count-based features (for instance for words found in the lexicons of two categories), the situation might occur more often than expected with either very short units of analysis or lexical resources of small size.

[8] A common point in terminology is that, for instance in supervised learning scenarios, the evaluation is performed as cross-*validation*—an evaluation that is based on several samplings on the given data set.

a crucial element in the validity assessment of automated content analysis, as it allows basic comparison and estimation of the evaluation scores, especially when the benchmark is geared to represent a specific variety of properties through the field of contestants.

To sum up, while the aspect of reproducibility of measurements for the same data set is an area where the automated versions of content analysis shine (as long as a deterministic paradigm is applied), most of our effort will go into sticking to the requirements of validity (especially external validity, i.e., generalizability (see (Neuendorf, 2016, p. 125)) where we conceive the held-out cases from an evaluation scheme as the points to measure this aspect). In parallel, we are aiming at several different other important qualities of the approach, such as transparency and inspectability as we will elaborate in the next section.

2.1.4 Further Desiderata

Having built a notion of validity and reliability as the most important two qualities of methodological instruments in the social sciences, we will now focus on some other traits and characteristics which will also play a role.

As we have laid out before, transparency is an important point from the view of social sciences[9]. But being transparent is not limited to being only a necessity in the justification process for the social sciences. Transparency is also beneficial for the development, understanding, and interpretation of any automated approach in the sense that it allows us to identify possible problems and address them in a purposeful way to mitigate the introduced error. Therefore, we consider transparency (at as many points in a pipeline of modules and in as many layers of modeling as possible) as the most important additional characteristic and therefore as an influential desideratum for our approach.

To accomplish the goal of building a transparent approach for the given problems, we will furthermore favor simplicity in the sense that we will not trade the central cornerstones of our strategy to achieve marginal improvements. Rather, we will attempt to improve the more simple approaches so that they are competitive with other ones. The only point where we think that simplicity can be neglegted is when the (accepted) complexity is encapsulated and hence leading to modularization. If we adhere to this exception, this allows also to replace the modules containing the complexity with more simple modules if desired. But in general, the requirement of simplicity will foster the overall transparency.

[9]Note that this is also closely linked to intersubjectivity, which refers to the famous concept of *Social Construction of Reality* (Berger and Luckmann, 1991) in social science, according to which there is no such thing as true objectivity and hence we cannot ask if something is true but if we agree that something is true (cf. Neuendorf (2016, p. 18))

The careful reader may already have noticed that we have repeatedly used terms like "pipeline" and "module" to point to another central characteristic: modularity. When we refer to modularity, we mean that we separate single steps (performed by a module) that have to be made to build the instruments to carry out automated content analysis. Hence, this allows for exchanging some of the modules (each of which carrying out a step in the pipeline) if desired.

The downside is that for such an exchange, the new module must adhere to the requirements of the other modules in the pipeline, i.e., it must be designed so that it accepts the same form of input and produces the same form of output as the module that we want to replace. This additional constraint does not have to be a handicap. If we choose the forms of the input and output so that human inspection is possible[10], we even accomplish one of the main goals: transparency. Furthermore, the possibility to exchange some of the modules also allows us to blend in already existing resources[11], be it through an (additional) external process or through the improvement of a single module in the pipeline. Modularity leads therefore to flexibility which in turn fosters the general applicability of the approach and thus should increase the chances of its adoption.

Last but not least, we would also argue that this kind of flexibility also increases the sustainability and maintainability of the approach. While it may be worthwhile to attempt to enhance single modules in order to improve the whole pipeline, it is easy to imagine that the adaptation for various specific use cases is accomplished by just changing one module (for example, plugging in different lexical resources, changing the underlying embedding model, or replace the classification component). This in turn allows one to re-use the pipeline for different purposes. In addition, there might also be good reasons to use only parts of the pipeline to produce resources which are then applied in different applications (e.g., applying the lexical resources in other software).

In other words, the modularity allows us to adapt the whole pipeline by changing only small parts of it but also single parts of it may be usefully applied if they deliver a valuable output on their own.

Using these two axes, we foster sustainability in at least two ways. First, because of the re-usability of the pipeline in many settings it is highly versatile. Second, the possible application of single modules leads to a freedom of recombination with other methods and software, so that the aspect of re-usability applies also to the components themselves. Additionally, these design principles are in line with the general claim for re-usability in

[10] If we follow that idea, this also clearly sets the goal to implement the approach so that inspection and interpretation is possible for every module in the pipeline.

[11] For example, lexical resources that are already available may be merged in or used as a basis from which we derive new ones

computational social science which has been made for example by Van Atteveldt et al. (2019).

As we have laid out the advantages of modularity, we consider this to be a strong argument to design the approach in this direction, especially since it allows us to link to the established methodology. At the same time, with a modularized approach, we propose an application of language technology that should be received as a complementary element or gradual shift in methodology instead of a replacement of paradigm.

We refer here to a paradigm in the sense that many exceptional achievements in recent research in computational linguistics are linked to the application of deep learning approaches (see Goldberg (2017) for an overview of methods). These approaches leverage the massive amounts of available (big) data and permit to solve the task at hand with much less specification of the *how*. In terms of machine learning based methodology, they are essentially bypassing the step of feature engineering. This step is the point where the researcher could integrate knowledge about the problem and domain which was assembled in a whole strand of empirical findings[12]. While on the one hand the application of deep learning approaches[13] has led to great improvements in many areas[14], some of the main downsides include the opaqueness of the resulting models (blackbox) (cf. Lipton (2018)), the demand for massive amounts of data to learn from, or the limited possibility for integration of prior knowledge (cf. Marcus (2018)).

Opaqueness is clearly an undesired feature if we strive for transparency[15]. The demand for larger amounts of data to learn from turns out to be problematic in the case of social sciences where we encounter many more empirical studies where the number of cases (e.g. articles, party manifestos, etc.)—or the amount of annotated data—is much lower[16]. For social scientists, it is also unsatisfactory if prior knowledge cannot be integrated into building the instruments for analysis (and the social scientist is instead reduced to delivering sufficient amounts of labeled data).

This is especially true for deductive workflows in research where hypotheses are derived from theories and in later steps conceptualized and operationalized to finally test them in an empirical study (cf. Neuendorf (2016)). While the ultimate goal of the social

[12]However, also linear models of machine learning (with engineered features) can be considered as a shift of paradigm to a largely data-driven methodology.

[13]Of course, also many earlier "classical" machine learning approaches (i.e., relying on linear models) were successful and are still competitive with deep learning approaches in some cases.

[14]We also point to the impressive improvements that have been made in a whole range of tasks since the advent of the transformer-based approaches like BERT (Devlin et al., 2019).

[15]The claim for transparency may refer to the level of the whole model, its parameters, or its algorithm (cf. Lipton (2018))

[16]Note that recently many researcher became aware of this problem and proposed different methods of transfer learning to reduce the needed amount of supervision. See Ruder (2019) for an excellent overview and Howard and Ruder (2018) for an application based study. However, transfer learning tackles mainly only one of the problematic aspects.

scientists as well as for computational linguists is to derive an instrument to measure (or predict) as precisely as possible (and therefore grant validity and reliability), it remains important for social scientists to be able to shape the instrument for the specific case at hand by integration of prior knowledge. This part of instrument adaptation may also be caused by certain criteria of replication and experimental settings from other studies in order to allow for comparability.

To sum up, we have given several reasons why modularity is an important feature of the proposed approach which contrasts with the goals of end-to-end solutions. Most importantly, it allows us to link to established methodology (dictionary-based methods), and enables us to follow a strategy where transparency (of the model, the parameters, and the algorithm) is a central pillar, which is linked to and ensured by other desired characteristics, such as simplicity, flexibility (or versatility) and hence sustainability. It is a major argument against the application of deep learning methods for our purposes, that those methods mostly rely on much larger amounts of training data than that which is available in the cases we are concerned with. While there is no reason not to recombine parts of the pipeline with deep learning approaches, we will not consider those settings for the sake of simplicity.

2.2 Derived Cornerstones

Based on the elaborated desiderata and criteria from the previous sections, we compile in this section a set of cornerstones for the design guidelines which we will take into account for this work.

First, we have considered some of the difficulties for interdisciplinary work and how to counteract them. While the most important point here is the acceptance in the target discipline, we also noted that the criteria from the source discipline play an important role which needs to be taken into account. We identified the need to comply especially with the basic requirements from the target discipline and intend to increase acceptability of the proposed approach through linking to established methodology accompanied by an automated solution for cumbersome steps therein.

Second, we have seen that reliability and validity are central methodological requirements for content analysis in a social scientist's view. Where certain aspects of reliability are easily satisfied through the (deterministic) automation itself, we will focus on other aspects through the choice of the empirical experimentation. The goal is that the approach itself works reliably for a range of (similar) cases and is not only suitable for a particular single case—a kind of assessment of external validity of the whole approach

(and not on the performance on a specific task.) In order to comply with the requirement of validity, we thoroughly evaluate the approach and compare it to other techniques in multiple settings.

As a side effect of complying with the standards of *validity* and *reliability*, our system needs to be designed so that it shows a certain *robustness*. On the one hand, we need to focus on the *generalizability* of the approach so that its applicability is given. In order to do this, we take challenging settings into account with respect to the available amount of labeled data as well as skewed distributions[17]. On the other hand, we also seek for robustness on the resource level. One way to accomplish this is by allowing for "imperfect" resources (e.g., that lexicons may be non-exhaustive and do not have to be disjunctive) and attempting to straighten out their disadvantages by other means. More precisely, we intend to create a solution which counteracts many of the standard problems for dictionary-based approaches like the lack of recall and ambiguity by an apt choice of modeling.

Third, we have derived additional desiderata that should allow us to satisfy these basic requirements while allowing us to pursue the goals we have set considering the strategy for interdisciplinary research. These additional desiderata are *transparency*, *simplicity*, and *modularity*. While we think that modularity is also fostering the *flexibility* and *versatility* of the approach, we see that those features in turn also contribute to *sustainability*. Additionally, the modularized design allows for the desired transparency, offering points for inspection (in the form of the resulting intermediate stages).

2.3 Chapter Summary

In this chapter, we have elaborated factors which foster successful interdisciplinary research, including the acceptance in the target domain.

To optimize incentives for adoption, we adapt to central requirements of the target domain and furthermore link to established methodology, namely dictionary-based content analysis for textual material.

First, we will propose a method to overcome a central problem for this area and facilitate the creation of lexical resources. The manual creation and curation of these resources tends to be costly. Nonetheless a demand for such resources exists, since available off-the-shelf resources often do not suffice (see (Grimmer and Stewart, 2013, p. 275), (Neuendorf, 2016, p.152f)).

[17] As noted in (Grimmer and Stewart, 2013, p. 276), this task is specifically hard, because "Supervised methods need enough information to learn the relationship between words and documents in *each* category of a coding scheme".

Second, we also propose a new way to apply these resources in order to improve the performance of the instrument we suggest to the social scientist. We therefore propose an embedding based modeling for textual classification integrating lexical-semantic resources.

Third, we develop the approach with respect to central guidelines derived in this chapter. Besides the general criteria for content analysis as a method (validity and reliability), we especially emphasize transparency which leads us to further rely on simplicity and, more architecturally, on modularity. This in turn allows us to also strive for flexibility and versatility, finally fostering the goal of sustainability.

Additionally, it has to be mentioned that these considerations do not come out of nowhere. They especially also emerged in the empirical reality of a larger research project (facing the double data skew, see Section 8.1.2 for a description) and relate heavily to the expected improvements regarding the modeling (generalization through semantically backed modeling).

To satisfy the demand for a valuable contribution in the source domain, this work can be seen as novel in two aspects. From the perspective of computational linguistics, it is an example of an alternative application of embeddings (which has been stipulated for example by Manning (2015)) for at least two different fields: lexicon induction and classification. Furthermore, the approach was specifically designed also for small-sized data sets for which deep learning approaches are not well suited. It also includes ways to tackle the imbalanced data distribution problem and allows applicability in a continuum from purely conceptual starting points (no annotated data) to purely data-driven approaches.

3

Related Work

> "If I have seen further it is by standing on ye sholders of Giants."
> — Isaac Newton, letter to Robert Hooke (15 February 1676)

```
In [180]: analogy(a="Steilpass", b= "Fussball",
x=None, y="Wissenschaft", model_given=model, verbose=True)

'Steilpass' is to 'Fussball' as 'Vorarbeit' is to 'Wissenschaft'

Out[180]:
[('Vorarbeit', 0.37803661823272705),
 ('Forschung', 0.3719194531440735),
 ('Zuspiel', 0.3677770495414734),
 ('Grundlagenforschung', 0.3584064245223999),
 ('Praktikern', 0.3565441370010376),
 ('Querpass', 0.3522380590438843),
 ('Wissenschafter', 0.34858861565589905),
 ('Abpraller', 0.3473680019378662),
 ('Akademie', 0.3449352979660034),
 ('Denkfabrik', 0.34179943799972534)]
```

In the previous chapter, we have elaborated central desiderata which we strive to take into account holistically when we develop our approach. As we have argued, we want to foster the acceptance in the target domain by following the path of an established

methodology. We have pointed out that the application of lexical-semantic resources (in the sense of "dictionaries"—lists of terms which are assembled to capture a concept or phenomenon) have a long history in social sciences.

As established as the method may be, we have also seen the demand to adapt or derive such resources (almost) from scratch to aptly perform automated analyses. This is costly in terms of invested effort—a drawback that is maybe not compensated for by the advantages of the approach, namely the ease of application and transparency in usage. Hence, we propose in this thesis a new way to use models of distributional semantics (embeddings) to derive such resources. In an additional step we will also demonstrate that the application of such resources can be coupled with an alternative modeling for text classification with remarkable benefits.

In order to provide the necessary background in this chapter, we turn to the related work in three fields which we integrate into the proposed approach. First, we consider models of distributional semantics and focus hereby on embeddings[1]. In this part we will also discuss the known shortcomings of these approaches and refer to work that tries to tackle those problems. Furthermore, we also present approaches that aim to encode larger units of texts (sentences, paragraphs, or documents) or that use the embeddings technique with a different goal in mind. Second, we will turn to lexicon induction and report on work that is directed to create lexical-semantic resources automatically. Third, we introduce briefly typical approaches to text classification as it is the underlying method for which we are going to propose a novel solution.

While these three fields may be viewed in isolation, we intend to combine different pieces from each field to form our novel approach. But since we believe that the separate treatment of the background for each field leads to more clarity, we have opted to keep the related work sections specialized for each of the three fields.

Nevertheless, we would like to emphasize that the different parts we combine (embeddings, lexical-semantic resources, and a classification component) are independent so that the freedom to exchange them on demand is granted[2]. In other words, the approach does not rely on a specific embedding model—as long as it provides us with the basic features of semantic vector space modeling. It also does not prevent the usage of any pre-existing lexical-semantic resource that could be helpfully integrated—as long as there exists a general embedding which covers a reasonable part of this resource and aligns with the incorporated semantics from the resource.

[1] We use the term "embeddings" since it was also used for such models as `word2vec` (Mikolov et al., 2013b) in the referring literature (see Levy et al. (2015))

[2] This follows our guideline of modularity which we see as a possible strength in the sense that also parts of the whole may be fruitful for other workflows.

Lastly, the classification component we propose (see Chapter 5) is not imposing any special restrictions on the two other components and may be swapped with any other classifier. The reader is therefore invited to see the related work section as a tour of possible alternatives rather than a prescription of hard requirements.

3.1 Embeddings

In this section we introduce models of distributional semantics and focus especially on embeddings. Since we intend to address different audiences with this work and for some of the readers the general concept of an embedding might be something new, we will give first an introductory whirlwind tour on what embeddings are and what they are good for.

With *embeddings* we refer to models of distributional semantics[3] which represent textual units (mostly words) as n-dimensional *vectors*.These vectors are calculated with the goal that textual units that carry the same meaning are represented by vectors which are similar to each other (see Sahlgren (2006), Turney and Pantel (2010), Erk (2012)). The vectors, and hence the models, are calculated by measuring the occurrence and co-occurrence of the textual units in a corpus (of texts). Furthermore, there are different *contexts* for which we can measure the occurrence of the units, such as a document, or a context-window around the unit, e.g., a certain number of words before and after a word which we want to model (cf. Turney and Pantel (2010)).

The measurement is often not the bare count of the occurrence, but rather a function that is related to a specific association, such as pointwise mutual information (Church and Hanks, 1990). Mostly, a transformation aiming at dimensionality reduction (e.g., SVD) is subsequently applied to map the vectors of the units into a lower-dimensional space (cf. Lowe (2001)).

While there exists a whole strand of research on different proposals to calculate such models (see Turney and Pantel (2010)), Mikolov et al. (2013a) proposed an efficient way to calculate embeddings with a shallow neural net and improved the approach in another paper (Mikolov et al., 2013b). This was accompanied by the public release of a large, pre-trained model and of the source code for its creation: `word2vec`[4]. In

[3]These models are also called *vector space models* (see Erk (2012)). The term *embedding* was introduced into the field by researchers who work mainly on deep learning approaches which are based on neural nets. The idea of an embedding layer (for neural nets) for language processing was introduced by Bengio et al. (2003), the term *embedding* was used in this way in Collobert and Weston (2008)

[4]The availability of a large, pre-trained model and of the source code have for sure contributed to the popularity of this approach. But also the efficiency with which other researchers were able to train on huge corpora of raw texts was key to its burst in diffusion.

contrast to the count-based methods, the architecture of this model was prediction-based and outperformed many state-of-the-art approaches on different tasks about synonymy, relatedness or analogy (see Baroni et al. (2014)).

One property that all of these models of distributional semantics share is that they are computed on large corpora of raw texts. The main idea behind the process of deriving semantical properties for textual units from their context is the distributional hypothesis (Harris, 1954). The hypothesis can be linked to the famous formulation by Firth (1957, p. 11) "You shall know a word by the company it keeps", which condenses the assumption that the contexts of a word (or its usage in language) defines its meaning. Theoretically, given a corpus that is large enough, one should be able to calculate the meaning (in relation to other words) for each word in the corpus.

As we have already mentioned, there are many ways to compute models of distributional semantics and there are also many ways to adapt their calculation to specific purposes. However, in this section, we do not focus on the plethora of different models from the perspective of their generation[5], although we give some background which should enable the reader to acquire a notion of it. Rather, we will take a perspective where our main interest is to find out how to use these models for solutions of the concrete problem at hand. Additionally, we would like to elaborate on the known shortcomings and point to further developments in this research area.

Note that we do not provide a detailed overview on the vast amount of literature on this topic since this lies beyond the scope of this work. For a general overview of the development in the area of distributional semantics, we recommend the article by Lenci (2018). For an in-depth study of count-based models, see Sahlgren (2006). For a systematic comparison study of newer models, see Levy et al. (2015).

3.1.1 Embeddings as Models for Distributional Semantics

In this section, we will briefly mention some of the remarkable properties of embeddings as models of distributional semantics that play a crucial role for the approach that we present in this thesis.

First, the most important property of such a model of distributional semantics is that it represents the vocabulary in a way which lets us identify terms that are similar in meaning. More formally, for any words w_u, w_v in the vocabulary V of the model, there are vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$ which represent these words in the vector space of dimensionality

[5]This is also because we do not propose any contribution to this field in this thesis. We *use* these models for specific tasks and perform only minimal adaptations to train them—following the guideline of simplicity. See Chapter 4 for a description of the model that we used.

d. This lets us compare the vectors and calculate their distance or their similarity, respectively. Mostly, the cosine similarity is used for this purpose[6].

$$\cos(\boldsymbol{u}, \boldsymbol{v}) = \frac{\boldsymbol{u} \cdot \boldsymbol{v}}{\|\boldsymbol{u}\|\|\boldsymbol{v}\|}$$

This measure (the cosine of the angle between the vectors) corresponds to the similarity of the meaning of the words, giving high values for vectors which have a similar direction. To find the most similar words for a given word, we compute these similarity values for all word combinations in the vocabulary. For example, in Table 3.1, we display the five most similar terms to *Computer*[7]. Note that we have a mixture of different forms of similarities: while *PC* (personal computer), *Laptop*, and *Rechner* may be considered as synonyms, *Software* is strongly related to *Computer* but certainly not a synonym. *Gerät (device)* on the other hand is a hypernym.

5 most similar entries to "Computer"		
Rank	Word	Similarity
1	PC	0.7978
2	Laptop	0.7643
3	Rechner	0.7456
4	Software	0.7098
5	Gerät	0.6991

TABLE 3.1: 5 most similar terms to *Computer* in the semantic space of the word2vec model, ordered by cosine similarity

While many applications in natural language processing (and especially those from the realm of deep learning) just use the embeddings as a basic transformation layer, an obvious case for their application appears to be lexical extension, i.e., finding more words that are of the same meaning or strongly related to the given word. In Chapter 4, we propose a method that relies on this central property of an embedding model.

A second interesting finding in Mikolov et al. (2013c) is that relations between word pairs were also usable to solve analogy questions such as *man* is to *king* as *woman* is to X (where the result for X should be *queen*) through the application of simple vector addition and subtraction (see Mikolov et al. (2013c) and Levy and Goldberg (2014a)

[6]Other distance measures like the Euclidean distance are also used. The cosine similarity has turned out to be practical (especially for normed vector spaces, i.e., where all vector have a length of 1, which leads to a simplified computation of the cosine similarity) and robust for a wide range of applications.

[7]Since we have lemmatized the texts on which this model was trained (e.g., singulars and plurals of the same word are replaced with the same lemma), the model is geared towards semantics. In fact, this model is not suitable for most of the syntactical similarity tasks proposed in Mikolov et al. (2013a) as most of this variability is stripped away in the preprocessing. On the other hand, this preprocessing helped to generalize better for semantics by reducing the variability of vocabulary which is not caused by a change of meaning, e.g., singular and plural forms of the same noun or different forms of a verb.

for more details on the calculation methods for these tasks)[8]. This means that the model captures these relations, e.g., *man:woman* (gender male:female), which can then be applied to other words like *king* to retrieve the analogous *queen*.

A further interesting property we like to leverage is the additive compositionality. In Mikolov et al. (2013b), the authors show that a simple vector addition of two words produces meaningful results in the sense that semantic properties of the vectors are combined[9]. For example, *Czech + currency* results in a vector that is closest to *koruna*, *German + airlines* leads to *airline Lufthansa*, and so on.

Although the authors do not evaluate this property systematically, it is promising in the sense that simple vector addition serves as a proxy to combine different semantic entities. We use this combinatory axis in our lexicon induction process as well (see Chapter 4 for the algorithm and Chapter 6 for empirical results).

While the close investigation of several different approaches to calculate embeddings (see next section) revealed that none of them is generally superior over all domains of application, there have been numerous proposals to improve embeddings. Besides the empirical experiments which aimed at scrutinizing those models in more detail, there was initially less focus on the question as to *why* these models work so well. More recently, there has been some progress concerning the explanation and the theoretical foundation of the properties outlined above (see Arora et al. (2016), Gittens et al. (2017), Ethayarajh et al. (2019)). However, the main impact that the introduction of those models had in the landscape of natural language processing was certainly based on its effectiveness in use (cf. Manning (2015)).

3.1.2 Approaches to Calculate Embeddings

In this section, we will focus on one specific algorithm to calculate embeddings. As we have pointed out beforehand, there is a whole strand of research consisting of different methods for calculating embeddings which we do not strive to cover in this section. Instead, we will briefly introduce the skip-gram model (with negative sampling) of `word2vec` to explain the general idea behind deriving a semantic model of language from a corpus of unlabeled text. We will not cover all the details of the model, especially

[8] In Mikolov et al. (2013c), the authors focus as well on syntactic relational patterns, like *small* is to *smallest* as *big* to *X*, with *X* being *biggest*, i.e., the superlative relation for an adjective. These syntactic analogy tasks are normally integrated in the evaluation battery for embeddings. However, since this kind of syntactical relations are not central to the application in this work, we will not further go into details in this direction.

[9] Technically, this means the element-wise addition of the two vectors and finding the closest vectors in the model for the resulting vector. Note that the resulting vector is not normed—which is not a problem for cosine similarities but potentially problematic for other similarity measures which are sensitive to vector length

leaving out the training mechanism itself[10] since we only want to provide a high-level understanding of the procedure. Rather, we will additionally point to some intriguing details which enhance the algorithm through a clever setting of default parameters (or hyperparameters as they are called in Levy et al. (2015)). In a second part we will refer to findings from scholars who have scrutinized different approaches to calculating word embeddings and summarize some of the important points in this line of research.

The core of the idea for models distributional semantics is, as mentioned beforehand, the distributional hypothesis (Harris, 1954, Firth, 1957), which states that the meaning of a word[11] is based on its contexts. Hence, such models rely on a certain notion of *context*. In the example that we show here, the context of a word is a set of words in its proximity. For instance, consider the following example sentence:

The brown dog barks and scratches its fur.

When we focus at the word *barks*, we would consider *The*, *brown*, and *dog* as left context of *barks*, while *and*, *scratches*, *its*, and *fur* are the right context.

Left context			**Word to model**	**Right context**			
The	brown	dog	barks	and	scratches	its	fur

Typically, the contexts which we take into account are limited to a certain number of units, which is often called the *window size*. For instance if we set the window size (ws) to 2, we would end up with

Left context, ws = 2		**Word to model**	**Right context, ws = 2**	
brown	dog	barks	and	scratches

Based on this context window, we count the co-occurrence of the words by sliding this window across all textual instances of the corpus. The intuition for collecting this kind of information is that if we want to model the meaning of *barks*, we rely on its context and correlate the word *barks* for example with *dog* and *scratches*.

But for the model we are going to look at in more detail, the *skip-gram* with *negative sampling*, the basic task is not calculate these counts of co-occurrence but rather, the model tries to predict the context of the words[12]

[10] This concerns the updating of the learned parameters using backpropagation with stochastic gradient descent.

[11] For the sake of simplicity we refer in this section to words as units of analysis although we use "term" for units which we want to embed in other parts of this work.

[12] The continuous bag of words (CBOW) of `word2vec` (cf. Mikolov et al. (2013a)) is another version where the model predicts the word of interest given its context. While this might seem to be more intuitive conceptually, the skip-gram model that we explain here instead has proven empirically to perform better—although CBOW has its advantages in some areas.

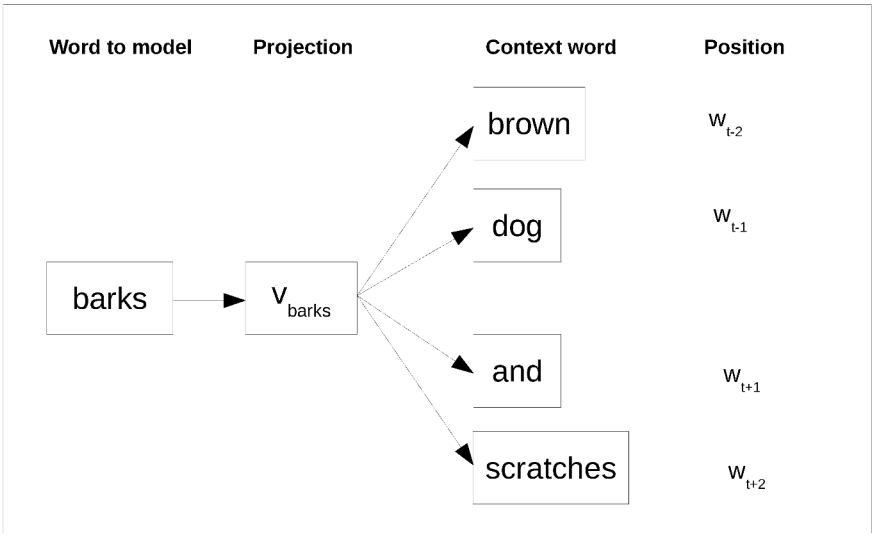


FIGURE 3.1: Basic architecture underlying the skip-gram model. The figure is based on Mikolov et al. (2013b) and is adapted to the concrete given example.

Consider Figure 3.1, which depicts schematically the architecture of the neural model. The algorithm performs the following: given that we have a word w_{t0} *barks* with a context (window size = 2), we are searching for a representation (a vector of dimensionality d) $\boldsymbol{v}_{barks}$ that is suitable to predict its context words *brown*, *dog*, *and*, and *scratches*.

Note that in the figure, besides the word *barks*, we have also put in a vector $\boldsymbol{v}_{barks}$ (the embedding) for the word *barks* while the target words are not depicted as vectors. This is because technically, in the model those are represented as one-hot vectors (which are not from the same vector realm as the embedding vector, i.e., in this schema the understanding should just be that we use a word to predict other words. But while performing this prediction, we use a projection that will in the end be our resulting vector).

With the given example, for one instance created by the sliding window (where *barks* is the center word), we get the word-content pairs (*barks*, *the*), (*barks*, *dog*), (*barks*, *and*), (*barks*, *scratches*). For each of these pairs we now attempt to predict the context word given the center word.

To perform this prediction, the skip gram-model represents each word $w \in V_w$ and each context (word) $c \in V_c$ as vector of d dimensions[13]. We would like to emphasize what

[13]These vectors are sometimes also referred to as two matrices W and C, where the rows in W are the word vectors, hence leading to a matrix of dimensionality $|V_w| \times d$. Accordingly, C is a matrix with context vectors as rows, leading to a matrix of dimensionality $|V_c| \times d$.

is often only implicitly mentioned in summarizations of the given model: for each word two vectors are learned: the word vector and the (word-as-)context vector[14].

In the version with negative sampling, a function of the dot product of these vectors $\boldsymbol{w} \cdot \boldsymbol{c}$ is maximized for (w, c) pairs in the corpus and minimized for false pairs (w, c_N) which do *not* occur in the corpus. These false pairs are constructed by stochastically corrupting other (w, c) pairs from the corpus (cf. Levy et al. (2015), p. 213).

In other words, the model is learning a representation for the word vectors (collected in the matrix W) and the content vectors (collected in a matrix C) while trying to accomplish the task of distinguishing between correct pairs (w, c) (if we take the above example, an instance would be (*barks*, *dog*)) and false pairs (w, c_N) (for the given example, this could be something like a false pair (*barks*, *pizza*) where *pizza* has been drawn as a negative sample). This representation will lead to word vectors which are similar for words that have a similar context—which is the essence of the distributional hypothesis.

The core idea behind negative sampling is to simplify the learning objective which would otherwise be computationally costly[15], i.e., proportional to the size of the vocabulary (which exceeds normally 10^5), concerning both the computation of the softmax and updating the weights of the output matrix.

While this general version of the algorithm leads to word vectors with the mentioned qualities, there is a number of additional settings which enhance the result of the calculation. According to the experiments conducted in Levy et al. (2015), at least the following hyperparameters are identified:

- *Dynamic Context Window*: Since words that are closer to the target word are considered to be more important, normally a weighting scheme (giving less weight to word-context pairs with larger distance) is applied. Technically, `word2vec` applies the weighting scheme via uniformly sampling the actual window size between 1 and the length of the window.

[14] This was made explicit in Goldberg and Levy (2014) where they relate this to the fact that if the word vector and the context vector were the same representation $\boldsymbol{v}_{word}$, there would arise an inherent problem: since the whole process is based on the conditional probabilities of the occurrence of context word c given word w, $p(c|w)$, and words only seldomly occur as their own context (the authors give the example of $p(dog|dog)$), this entails that the model should assign a low value to $\boldsymbol{v}_{word} \cdot \boldsymbol{v}_{word}$, which is not possible.

[15] This cost comes from the softmax calculation for which we would need to calculate the sum of the dot products between the center word vector and *all* other contexts. Mikolov et al. (2013b) have proposed hierarchical softmax and negative sampling as two strategies to reduce computational costs. Negative sampling can be seen as a simplified proxy for noise contrastive estimation, and since we hence neglect to really approximate the log probability of the softmax, we may use a k number of negative samples to learn the distinguish the target from noise using logistic regression. The interested reader may refer to the original paper (Mikolov et al. (2013b)) for more details and to Levy and Goldberg (2014b) for and in-depth analysis how this method is related to matrix factorization as these parts exceed the scope of this work

- *Subsampling*: This step removes highly frequent words which are supposed to be non-informative or less informative (i.e., stop words) according to a given threshold and probability scheme (Mikolov et al., 2013a). Since `word2vec` removes those words before the pairs from the corpus are created, this is actually increasing the real window size in the sense that the window size *per se* remains constant while the removal of the words from the original text leads to a larger text span been covered by the window size.

- *Deleting Rare Words*: Words that do not occur enough in the corpus to meet a certain threshold are not taken into account. Since their deletion is also performed before the context window calculation is applied, this influences the observed word-context pair generation as subsampling does which is described in the last point.

- *Context Distribution Smoothing*: Negative sampling requires a distribution according to which the samples are drawn. Mikolov et al. (2013b) report that a smoothing of the unigram distribution by raising the counts to the power of a constant (empirically 0.75 turned out to work well) leads to better results than the unigram distribution. This is intuitively justified as more frequent words are downsampled in this way before the negative sampling takes place.

- *Number of Negative Samples*: The number of negative samples increases the quality of the estimation (based on the differentiation between the real pairs and the false pairs (negative samples)). In the end, this has an influence on the embedding quality as well. In general, a rule of thumb is that the larger the corpus is, the lower the number of negative samples may be set.

Since Mikolov et al. (2013b) have demonstrated such remarkable features of their model, like being able to resolve analogy tasks in an unsupervised fashion with simple vector offset methods, and to perform basic semantic composition (alongside its main capability of modeling meaning in the vector space leading to relative similarity or closeness in space for units with similar meaning), their `word2vec` model was widely embraced[16].

Baroni et al. (2014) measured the performance of prediction-based models against count-based models which revealed that the former outperformed the latter in a broad set of empirical evaluations, hence also the name of their paper "Don't count, predict!".

Pennington et al. (2014) proposed another model, `GloVe`, where the vectors are learned from a co-occurrence matrix in a log-bilinear regression. The authors perform various benchmarks (including similarity tests, analogy tasks, and downstream applications like named entity recognition) to claim the superiority of their model.

[16] Also the efficient implementation of the software that enabled other scholars to calculate such embeddings for different applications and experiments helped to foster its popularity

After having scrutinized `word2vec`'s skip-gram model in Goldberg and Levy (2014) and pointing out the inherent closeness of neural word embeddings and matrix factorization methods[17] in Levy and Goldberg (2014b), Levy and Goldberg (2014a) also investigated the seemingly large gap between neural models and count-based embeddings for the similarity and analogy tasks.

Finally, this "battle of the models" culminated in the thorough empirical evaluation of the different approaches in Levy et al. (2015), focusing on the quest to reveal what is leading to the measurable differences in evaluation tasks, given that the models all appeared to be related to each other. In their paper, Levy et al. (2015) point out that the most important differences are the hyperparameters of the models. In their extensive experimental setup, they compare explicit (i.e., PPMI matrix-based) embeddings (with and without SVD), `word2vec` embeddings (skip-gram with negative sampling; SGNS), and `GloVe` embeddings and evaluate them on similarity and analogy tasks. The main finding is that the "natural" (i.e., per definition or per default) choice and tuning of different hyperparameters is the main cause for the difference in their performance, and that, as a consequence, the algorithms perform roughly on-par if they are aptly tuned.

Hence, in their re-evaluation of prior claims they conclude that prediction-based models do not outperform count-based methods if the same tuning is applied to the latter. They also decline that `GloVe` vectors are better than `word2vec` vectors and actually find the inverted result in their experiments. Also the advantage of PPMI models against SGNS in analogy tasks, as claimed in Levy and Goldberg (2014a), vanishes and the experiments reveal that SGNS captures some syntactic analogies better.

Additionally, the slightly modified formula for similarities, using cosine similarity multiplication instead of addition (as proposed in Levy and Goldberg (2014a)), produces better results, especially for the count-based methods which benefit the most from this modification. Summarizing their findings, the authors give a list of harmful and beneficial configurations and provide practical recommendations. Among them is also the conclusion that the SGNS variant of `word2vec` is a robust baseline for most tasks, the fastest one to train, and has the lowest computational requirements, which may also play an important role.

Mikolov et al. (2018) also summarize findings concerning specific enhancements for embedding calculation—or pre-training of distributed word representations. In addition to several other settings that we already mentioned in the list of the "hyperparameters", they point to the performance-wise positive contribution of pre-computed phrases (i.e.,

[17] They show that the skip-gram with negative sampling is implicitly factorizing a word-context PMI (point-wise mutual information) matrix

multi-word terms), as in Mikolov et al. (2013b)[18]. Also the usage of subword information, as applied in `fastText` (Bojanowski et al., 2017), may contribute to an improvement and specifically tackles the problems of rare or misspelled words and offers a possibly better solution for morphologically rich languages.

Another interesting empirical setup is presented by Sahlgren and Lenci (2016) where they investigate specifically the performance of different models of distributional semantics when these models are calculated on limited data[19]. The main finding is that the neural models (in this experiment the models from `word2vec`) perform worse with smaller corpora than their count-based counterparts and the model based on inverted (truncated) SVD[20] performs surprisingly well.

Additionally, they also evaluate the quality of the representation of words according to their frequency. As expected, words with medium-to-high frequency are better modeled, although it is interesting that the count-based approaches model words with medium frequency better than words with high frequency—in contrast to neural models and inverted SVD which perform best for words with high frequencies. The performance for low-frequent items faces a drop for all of the tested approaches.

A noteworthy approach to determine the best unit to embed in a data-driven fashion is presented in Gyllensten et al. (2019) where the authors attempt to infer with a recursive algorithm what the "tokens" are that they want to embed. In this way, they include embeddings on the subword level (from character sequences to syllables), as well as multi-word terms and even larger units which span across fossilized textual sequences like collocations and stereotypical expressions.

3.1.3 Known Shortcomings and Possible Remedies

While the resulting embedding models are performing partly in an astonishing way, we must also be attentive to the shortcomings and drawbacks of such models. In this section, we will therefore discuss some of the known problems in more detail.

[18] It might be important to know that not all of the identified instances of the phrases get converted to atomic terms (normally, they will be glued together with a "_" so that the tokenizer considers them as one word), but only half of all cases.

[19] In order to show the deterioration of the performance of the models with respect to decreasing amounts of data, they shrink the corpus they train on from 1 billion to 100 million, 10 million, and 1 million words. For the setting with only one million words, the models barely match the random guess performance, if at all.

[20] This is a factorized matrix model in which the result of a singular value decomposition is not treated in the "normal" way where one keeps the first n dimensions of the resulting matrix. Instead, the first n dimensions are removed.

Out-of-vocabulary Words and Rare Words

One of the obvious problems for models that are based on a limited static vocabulary is that there is no representation for certain terms—they are out-of-vocabulary (OOV). There are several reasons why there would be no representation for some terms.

First, the term could not be present in the corpus at all, which obviously leads to the lack of representation in the models. This problem is aggravated by the fact that for computational feasibility and quality of the model, we normally shrink the vocabulary of terms for which we create a representation to a tractable size, ignoring terms which do not occur often enough to meet a given threshold.[21] Second, the term which we want to embed[22] might come from a domain where there is less adherence to canonical spelling like in social media. In such domains, transformation processes like abbreviation of phrases ("lol", "lmao") or lengthening ("coooollllll", see Brody and Diakopoulos (2011)) are undoubtedly pervasive. Also social and geographical influences on the more informal language as well as illiteracy are further causes for non-canonical language (see Eisenstein (2013) for an overview).

There are two traditional ways to mitigate the impact of this variance. One may either adapt the resources (in this case the embedding) to the textual source (which would mean creating specific models for such domains), or one attempts to normalize the language, which means applying preprocessing on the raw text that maps it to a more canonical form.

Since the retraining of an embedding is costly and domain adaptation is an art in its own right, there were several proposals for how to tackle the out-of-vocabulary problem for embedding models. Bojanowski et al. (2017) proposed `fastText`, a model that enriches the word embedding model with subword information. Subword information, in this case, means that in addition to the encountered word forms, their decomposition into character n-grams also is integrated into the learning algorithm. More technically, the set of character n-grams are summed up (as well as a separate vector representation of the full word) to learn its embedding given the context[23]. While one idea is to capture further semantic information that is linked to morphological processes (e.g., prefixing or

[21] This is especially important for languages with a rich morphology like Finnish or Turkish, where these linguistic processes lead to many possible word surface forms that are very rare. Also other productive language processes such as compounding in German that increase dramatically the list of terms for which we need a representation. Furthermore, even drastically enlarging the covered vocabulary by lowering the threshold would not suffice to solve this problem because the given corpus would not comprise of all compounds that might be built in the future by no means.

[22] This is the process to retrieve a vector representation in the model.

[23] Note that the idea to represent a token with subword units (based on byte-pair encoding) for rarely seen words was also introduced for neural machine translation by Sennrich et al. (2016)

suffixing), one of the main advantages is that if we encounter an OOV term, we may fall back to a vector representation based on the sum of its character n-grams.

This solves the OOV problem in principle, given that the unit we want to embed is at least composed of character n-grams included in the model[24]. The main intention of the authors was to create meaningful (or better) representations for unseen (or rare) words—especially for morphologically rich languages—which they demonstrate on data sets for four languages.

Character-based modeling for embeddings are also found in other approaches, such as the CHARAGRAM model from Wieting et al. (2016a). In contrast to the `fastText` model, the authors focus here on training task-specific resources which make use of the character-based representation of input sequences.

One of the advantages of embedding models such as `word2vec` (Mikolov et al., 2013b) or `GloVe` (Pennington et al., 2014) is that large pre-trained models are readily available. But these models do not provide a method for OOV words. In order to address this problem, Patel et al. (2018) have presented an approach to add a similar functionality ex-post, including awareness of morphology-based variability as well as transforming processes like lengthening.

Another line of research goes in a different direction. Besides leveraging subword information or creating character-based models one could also stay closer to the idea that inferring the meaning of an unseen term (or more practically: unrepresented term) should also be possible with only a few instances (given that the surrounding context of the unseen term is understood).

This application of the distributional hypothesis is much closer to how humans would try to infer the meaning of an unknown word. Imagine that one needs to estimate the meaning of the masked word XYZ in *"The XYZ barks and scratches its fur."*. If one has already the knowledge about the world and knows what *barks* and possibly *scratches its fur*, it is easy for us humans to guess that XYZ is something like a dog—only after having seen this single example.

Khodak et al. (2018) strive to accomplish this task and present their *à la carte embedding*. The main idea is that there is a projection from the context vectors of the words to the embedding vectors which can be learned from the given embedding[25]. This linear projection is further applied to the given instances of contexts surrounding the unseen word in order to create a new embedding vector.

[24] Of course, a further fallback could be to use only the character n-grams for which there is a representation as an approximation if some of the character n-grams are not included in the model.

[25] This is motivated by the findings of Arora et al. (2018).

More concretely, one uses the instance(s) of the unseen word with its context(s) and calculates an ad-hoc vector for this word that approximates its value in relation to the given embedding model. The advantage of this model is that it is also applicable to generate embeddings for multi-word terms and idiomatic or entity-related expressions, since it relies only on the contexts of the given instances. Consequently, in contrast to the above mentioned methods which attempt to capture and leverage subword information to reconstruct a word vector, this method needs to have access to the instances of the unseen word (e.g., the sentence containing it).

Polysemy

An additional consequence of the static modeling of terms in embedding models is that surface forms which have several meanings or senses (e.g., *bank* as financial institution and *bank* as side/edge of a river), have only one representation in the embedding. This is also called the meaning conflation deficiency (Camacho-Collados and Pilehvar, 2018, p. 747).

Some researchers propose to tackle this problem by generating a separate representation for each sense. This links to the task of word sense discrimination (cf. Schütze (1998)) where the different senses have to be identified and discriminated. Reisinger and Mooney (2010) cluster the contexts of the word occurrences to identify the different senses and use the centroids from each cluster as a prototype of the respective sense. Similarly, Huang et al. (2012) train a neural network for embeddings based on clustered contexts to keep the representations of different senses apart. To cluster the contexts they use a previously trained word embedding model without multiple types.

Instead of unsupervised clustering methods, Trask et al. (2015) concentrate on supervised scenarios. For example, they use a part-of-speech tagger to keep the senses with different tags apart. Consider that *"apple"* may be a fruit (which is detected by the noun tag "NOUN" from the tagger) or a company (which is tagged as a proper noun "PROPN" by the tagger). After having preprocessed the corpus in this way, they calculate the embedding based on these senses. They show that supervision applied in this fashion also works to calculate sarcastic meanings of adjectives in sentiment tasks. In a similar way, they leverage a named entity tagger to disambiguate senses of named entities ("Washington" as a name of a person vs. as a name of a state).

However, despite the critique on the single representation strategy in common embedding models, there were also arguments which lead in another direction. Gyllensten and Sahlgren (2015) argue that the conflation of senses is a misinterpretation due to the typical inspection of the vicinity of a word vector for which one normally concentrates on

k nearest neighbors (with a small k in order to keep manual inspection tractable). They argue that the different senses are still present in the model but they are intermixed, i.e., the vector is actually a composition of the different senses (weighted by their relative frequency). Furthermore, they show that if one increases k to a sufficient large number, one can leverage the structure of the network of nearest neighbors.

For example, the word *suit* has different senses, amongst them the most important are the *law*-sense (*lawsuit*) and the *clothes*-sense. While this leads to an intermixed list of nearest neighbors of the vector of *suit*, the different senses are recoverable because the nearest neighbors (e.g., case vs. jacket) belonging to each respective sense are not similar to each other. Following this idea, they use a graph modeling (Relative Neighborhood Graphs) to detect senses based on the local structures of the neighborhood.

Similarly, Arora et al. (2018) investigate the assumption that a word vector is a composition of its different senses and present another way to derive those different senses. They refer to a generative model with discourse vectors (Arora et al., 2016) to induce a sparse coding[26]. With this coding, they determine the different senses in the word embeddings which turn out to be usable for a competitive word sense induction procedure. This work is especially interesting since it relies on the linearity assertion which is theoretically justified as well as empirically tested. In simpler words, the embedding of a word is inferred through a linear projection of the embedding of its contexts[27] and given the model with the atoms of discourse, one can induce the senses.

The main difference between the latter two approaches and the others which were mentioned beforehand is that they reconstruct the different senses from a given embedding, in contrast to integrate external supervision (by hard-coded information or clustering induced structure) during the embedding calculation. On the one hand, it appears to be worthwhile to leverage already available semantic resources and integrate them into the process of the embedding calculation[28]. On the other hand, it is promising to pursue the ideas for models which enable inference of different senses ex-post.

In the end, it may lie in the eye of the beholder which approach is more feasible to detect and encode the different senses in the embedding on the modeling level for an

[26] More precisely, a discourse vector is a linear combination of atoms of discourse. Empirically, they determine approximately 2000 atoms of discourse to be sufficient to code the different discourses which are in turn the device to create the different senses. The authors stress the point that the linear combination of a small number of discourse atoms is the linear algebraic analog to overlapping clustering.

[27] They use the smooth inverse frequency (SIF) modeling—actually a proposed sentence embedding from Arora et al. (2017)—for the contexts.

[28] A recent encompassing survey on the topic of modeling senses in embedding methods is given in Camacho-Collados and Pilehvar (2018).

application[29]. If more control is required and resources for the concrete discrimination of senses are already available, one should definitely make use of these resources, maybe even in an imposed re-modeling step, e.g., with the techniques of retro-fitting (Faruqui et al., 2015).

Relatedness vs. Similarity

If we use embeddings as backing models of language, we should note that there is a mixture of several types of similarity. The most important distinction is the one between relatedness and similarity. Kiela et al. (2015) point out that two things may be semantically related, like *car* and *petrol*, while they are not similar in the sense that they mean the same thing. The authors lay out further that this is, on the one hand, not a new insight (referring to literature of the cognitive sciences (Tversky, 1977)) and, on the other hand, that this is an outcome of the distributional hypothesis itself because *car* and *petrol* may well occur in the same contexts, although not having the same meaning.

Since most of the embedding models learn both similarity and relatedness indiscriminately, we have to be aware that we get intermixed representations. While this may not conflict with the purpose of application of the model in some cases (in the lexical extension method shown in Chapter 4, we even make explicit use of this property), this basic distinction is potentially important for downstream applications. Hence Kiela et al. (2015) carry out experiments to compare methods of joint learning and retrofitting (Faruqui et al., 2015) to specialize embeddings to focus either on similarity or relatedness. They also demonstrate improvements on downstream task which rely on a distinction between relatedness and similarity.

From Source Bias to Model Bias

As embedding models are trained on massive amounts of raw text, the model of meaning that is derived relies on the *content* of the corpus. Bolukbasi et al. (2016) found that embeddings like `word2vec` and `GloVe` contain gender biases, supported by findings of other scholars (Caliskan et al., 2017). If we remind ourselves of the fact that the calculation of embeddings as models of distributional semantics is a heavily data-driven process, this is not surprising at all. The reflection of a bias in the embedding is in fact an *intended* outcome of this process, since we do not derive truth but a model of meaning relative to the given content, i.e., the collection of texts.

[29]Of course, the usage of contextualized embeddings, such as ELMo (Peters et al., 2018), or BERT (Devlin et al., 2019), is another strategy to cope with this problem. But since these approaches calculate the embeddings dynamically given the context of the instance, this raises the non-trivial question of how we would compare them to static resources such as dictionaries and word lists.

Nevertheless, one should be aware that these models reflect the bias of the corpus they were trained on. Hence we must take this into account if the performance of the downstream application is possibly sensitive to such a bias. Fortunately, there are also methods to counteract this bias, as presented in Bolukbasi et al. (2016) and also applied for publicly available embedding models like ConceptNet (Speer et al., 2017) where bias for gender, religion, or ethnicity is reduced.[30]

3.1.4 Embeddings for Sentences

As we have seen, we get a remarkably apt model of the semantics (concerning similarity) of the terms that are embedded. While the word-level semantics provide already a formidable axis to apply such models for natural language processing, one might also ask, how such word-level embeddings can be used to represent phrases, sentences, or larger textual units[31].

There are several attempts to produce such embeddings. Le and Mikolov (2014) proposed a way to learn embeddings for paragraphs and even documents. They relate their methods to the ones that were introduced by Mikolov et al. (2013b) in the sense that they frame the representation calculation as a prediction task. They present two models to create paragraph vectors. In the first model (called PV-DM), they introduce a paragraph token that is used in combination with contexts (from sliding windows over the paragraph) to predict the next word in the paragraph (the paragraph token vector is concatenated with the context vector for the prediction task). While the word embeddings are shared across all paragraphs, the paragraph vectors are only shared for the prediction within the paragraph. The second model (called PV-DBOW) that they propose is similar to the skip-gram architecture and tries to predict words of the paragraph (ignoring word order) using the paragraph token vector. In order to reach the best performance, they concatenate the paragraph vectors from both models. As one of the main goals of this approach is to learn fixed-sized embeddings (vectors) for textual units of variable length, this approach is of course applicable to single sentences.

While the idea to leverage the same core algorithm as for word embedding inference worked reasonably well, other model architectures have also been explored, especially such ones that use the encoder-decoder architecture. In encoder-decoder models, a neural network is trained to encode the unit of interest and a decoder is trained to

[30] See `https://blog.conceptnet.io/posts/2017/conceptnet-numberbatch-17-04-better-less-stereotypedword-vectors/` for a more detailed description.

[31] There are also many approaches that embed larger textual units directly, i.e., without using a word embedding model. However, since we focus on the possibilities to use the word embeddings to create representations for the larger units, we do not revisit them in detail.

predict a certain target variable from the encoded state. If the target to predict is the original input (often a sequence), this is an auto-encoder.

In a kind of combination of the encoder-decoder model with the idea of the skip-gram, Kiros et al. (2015) learn a model in which a sentence is used to predict the sentences around it. This is in close relation to the skip-gram architecture for word embeddings (hence the name "Skip-Thought")[32].

Adi et al. (2016) (and in a follow-up Adi et al. (2017)) have compared methods of sentence embeddings and used the prediction of length, order and content as a proxy for an evaluation. They find that the baseline of just averaging word vectors of the sentence performs well for the task of content prediction, i.e., learning to predict if a word is present in a sentence or not. While for length (number of words in a sentence) and order (of given words) the LSTM-based (long short-term memory (Hochreiter and Schmidhuber, 1997)) encoder models perform better, the simple averaging models perform remarkably well in the content task where they even outperform the models based on the sentence embeddings from the encoder-decoder learned model. Also Wieting et al. (2016b) find sentence embeddings based on averaging (of word embeddings), which are trained for the task of sentence similarity prediction[33], to work impressingly well, especially for out-of-domain data. They are also competitive or even superior for sentiment analysis and entailment prediction tasks.

Arora et al. (2017) propose another sentence embedding scheme which is in contrast to Wieting et al. (2016b) unsupervised. Their approach is based on *SIF* (smooth inverse frequency) where the word embeddings are weighted for averaging by a factor relative to the word's frequency. Additionally, they also remove the projection on the first principal components of the sentence embedding. They show that their unsupervised weighting scheme produces sentence embeddings which allow them to outperform other supervised approaches in the sentence similarity task which was used in Wieting et al. (2016b).

Pagliardini et al. (2018) also propose a fast method to produce sentence embeddings. Their method scores better than SIF embeddings (which use a static weighting) for supervised tasks but is beaten on unsupervised tasks, hence, once again, pointing towards the powerful, simple (weighted) averaging method for unsupervised application.

`InferSent` by Conneau et al. (2017) uses a training set for natural language inference to learn to produce sentence embeddings. The authors demonstrate that these sentence

[32] Additionally, they use a method to enlarge the vocabulary where they project `word2vec` vectors into the embedding space of the sentence embeddings.

[33] Here, the word embedding used to compose the averaged sentence embedding are PARAGRAM-PHRASE embeddings. Note that these word embeddings are trained for the task at hand and averages over pre-trained embeddings.

embeddings are better suited to learn classifiers for several prediction tasks than other methods such as Skip-Thought embeddings, proposed by Kiros et al. (2015).

However, as shown again in Ettinger et al. (2018)[34], the averaging of word vectors constantly performs well for tasks which are mainly connected to the content of the sentence (and less dependent on the order of the words.) Similarly, also Conneau et al. (2018) find that a Bag-of-Vectors sentence embedding performs surprisingly well and confirm this finding for the word content (WC) probing.

More recently, Shen et al. (2018) showed that the average-based composition of word embeddings in order to embed a sequence is improved through simple max-pooling (i.e., taking the maximum value from each dimension in the word vectors) or concatenation of the max-pooled version with the averaged version. This simple scheme outperforms much more complicated sequence embedding models learned by several different neural net architectures.

Additionally, they show that tasks which are more sensitive to word order, like sentiment detection, profit from a hierarchical pooling. More precisely, this pooling is based on a first step of local averaging followed by a step of global max-pooling. Although their methods perform better on document-level tasks (like document categorization and sentiment analysis), they are also applicable to shorter sequences like sentences or paragraphs.

To sum up, there are many methods which either make use of supervision to train an embedding function for sentences or apply other "unsupervised"[35] methods like `Skip-Thought`. While these representations may be useful for modeling the textual input for several downstream tasks, they do not constantly outperform simpler baselines like SIF sentence embedding or even averaging. An additional downside of these learned projections for sentences is that they are not directly comparable to word vectors in the same space. Since we aim in this thesis to follow the line of lexical resources and strive to apply them in an embedded modeling, we take the good results concerning the content predicting task as a promising indicator and thus rely on a simplistic model for sentence embeddings.

[34] While the authors are not mainly interested in benchmarking the sentence embeddings *per se* but rather to assess them with respect to specific qualities, they used the averaged vector of a sentence as a baseline which almost scored perfectly on the content probe task.

[35] In fact, the `Skip-Thought` model is supervised in the sense that the sentence before and afterwards are given. Of course, it is not difficult to assemble a corpus of sequences of sentences. But it restricts the training to corpora consisting of texts with ordered sentences.

3.1.5 Summary

We have given a short introduction to models of distributional semantics and especially on embeddings. Models of distributional semantics incorporate representations for terms which are calculated through an application of the distributional hypothesis, stating that words are defined by their context.

Given such a model of distributional semantics, we determine how similar words are to each other. Using an embedding model, for instance computed with `word2vec`, we retrieve dense vector representations with which we can also perform other comparison operations, e.g., analogy tasks or meaningful vector addition.

Note that the presented approach in this work does not settle for a specific flavor of the model. On the contrary, it is created to foster the possibility to exchange different pre-trained models. More precisely, although we use a `word2vec` model for the experiments in this thesis, exchanging the underlying embedding model is generally not restricted [36].

This means that we may also apply different embedding models for different purposes. While the modeling of specific terms may be more apt in a certain pre-computed model, the user could also switch to a different model, let us suppose a `fastText` embedding, to profit from the benefits of subword modeling for languages with richer morphology.

3.2 Lexical Resources for Content Analysis

We have chosen to implement an approach which is linked to the established method of dictionary-based (automated) content analysis. One trivial requirement that follows from this choice is the availability of apt lexical resources, i.e., a list of words and their attributions to a certain concept.

With *concept* we refer in this work to an abstract idea which underlies the axis of comparison for the content analysis at hand. For example, we might be interested in the tonality of parliamentary discussions about a specific topic. Since we further would like to classify the content along an axis of tonality between negativity and positivity (i.e., along a scale), we operationalize the analysis by measuring both of them. Hence, the concepts we need to measure are positivity and negativity.

In order to measure the presence of those concepts in the content (i.e. the transcriptions of the parliamentary discussions), we use lists of lexical items that are semantically

[36] Of course, we cannot freely intermix the representations during one stage of application, but for instance the lexicon induction is completely decoupled from the classification with respect to the used embedding

linked to the concepts. For example the words *catastrophic*, *horrible*, and *failure* are linked to negativity. In the same sense *wonderful*, *fantastic*, and *success* are linked to positivity.

In classical dictionary-based approaches the occurrence of terms (in the unit of analysis) from those lists serves us as a proxy to measure the presence of the respective concept. We will call these lists "lexicons" in this work. A *lexicon* is thus a collection of terms which serves the purpose to detect a specific concept in a textual unit of analysis.

In contrast to classical approaches we propose to use a model of meaning, i.e., a word embedding, to represent the unit of analysis as well as the concepts that are represented by a list of terms. This allow us to conduct the comparison (or measurement) in the semantic space (see Chapter 5)[37].

In resources like the General Inquirer (Stone et al., 1966) or LIWC (Pennebaker et al., 2001), words are annotated as belonging to a concept (e.g., positivity, negativity, legal matters, military matters, and so on[38]) or not. In LIWC, there are also more linguistically motivated categories like part-of-speech (e.g., prepositions, articles, and so on), psychological constructs (e.g., affective processes or sensory processes) or personal concerns (e.g., work, money, religion, death) (see Neuendorf (2016, p. 151) for a summarizing description or Pennebaker et al. (2015) for a more comprehensive catalogue of dimensions).

We consider such resources as a compilation of a multitude of concepts. An application may be designed to take any number of the given categories from the lexical resource into consideration. In contrast to this, we will focus, on the one hand, only on the categories/concepts—and hence their manifestations as a list of words or terms—that we need (or which turn out to be at least helpful) to solve a task. On the other hand, we would like to have the freedom to create them on demand, extend them and adapt them for certain domains. Hence, we aim at developing an approach that satisfies the described catalogue of desired features.

The manual creation, curation, and adaptation of lexical resources is considered a time-consuming step in traditional computer-assisted content analysis (CATA) approaches. Neuendorf (2016) mentions the need to create apt customized resources while describing the process to compile these custom resources as "arduous" (p. 149) despite the features

[37] This enables us to generalize the lexical resource (in the sense that we compare on the level of meaning and not on the level of string matching). Additionally, we also address the problem of ambiguity of single entries in the lexicon: on the one hand, through the integration of context by modelling the text (the unit of analysis) on sentence-level. On the other hand, through the clustering and quantization of the lexical resources (see Chapter 5 for the description of this process and Chapter 6 for more detailed examples of such lexical resources.)

[38] see `http://www.wjh.harvard.edu/~inquirer/homecat.htm` for an overview of the 192 concepts assembled in the General Inquirer

of CATA programs like WordStat. Therefore, we will automate this step as far as possible and facilitate it (see Chapter 4).

We would like to mention here, that our understanding of a "lexicon" is intentionally a simple one: we do not aim to attribute more information to a term than simply being a member of a set[39] of terms that are related to a concept. In other words, we consider the lexicon to be a more or less exhaustively assembled collection of terms which relate to a (semantically based) concept. These terms may be words, multi-word expressions, any other helpful symbol, or a mixture of them. If we have to distinguish between different concepts, we would derive a lexical resource for each concept. If we only have to detect a single concept (or estimate the presence or absence of it) we aim to provide a way to measure the signal of the concept and its interpretation. In any case, we would refer to the entirety of employed lexicons that capture the concepts under investigation with "lexical resources"[40].

In order to put our approach in relation to other work in this direction (lexicon induction), we will introduce a number of other approaches, many of them connected to sentiment analysis, since this is a task where lexical resources are often applied (cf. Taboada et al. (2011)).

3.2.1 Inducing Lexicons

We consider the induction of lexicons as the task to create a lexical resource as a collection of lexical items which is aimed at being useful to perform other tasks, such as classification, sentiment analysis, or calculating descriptive statistics about a corpus.

In order to refer to the central literature in this field, we often get in touch with sentiment analysis, since this task is often carried out with the help of sentiment lexicons. Although this coupling of lexicon induction with sentiment analysis is not mandatory at all, we will include many examples from this field. However, we would like to point out that the approach we describe in Chapter 4 is by no means restricted to sentiment lexicons.

Additionally, in contrast to many of the methods that we mention below, our approach is not even relying on a semantic axis between two extremal points (such as positivity and negativity) and is geared to be flexible enough to perform creation, extension, and adaptation (see Chapter 6).

The induction or expansion of sentiment lexicons has been investigated in many variations from different perspectives and for various goals. Liu and Zhang (2012) divide

[39] We will also use the term "list of terms" interchangeably for "set of terms".
[40] Note that we also subsume subtly differently defined resources which are often named "dictionary".

in their survey the work in this field into three groups: manual approaches, dictionary-based approaches and corpus-based approaches. While the manual approach is time-consuming, it is still often used to create core lexicons which are not domain-specific, e.g., as in Taboada et al. (2011). Riloff and Wiebe (2003) produced another widely-known manually curated lexicon which was applied for sentiment analysis, for example in the OpinionFinder (Wilson et al., 2005). This lexicon was also used in many papers to create a baseline for comparison. As noted by Grimmer and Stewart (2013), these lexicons often also contain an estimation of the intensity for the respective modeled dimensions—positivity and/or negativity for sentiment analysis purposes.

The dictionary-based approaches (which are sometimes called thesaurus-based approaches as in Huang et al. (2014)) utilize pre-existing dictionaries or thesauri like WordNet (see for example Kim and Hovy (2004), Esuli and Sebastiani (2006), Baccianella et al. (2010), Neviarouskaya et al. (2011)). Note that the term "dictionary" refers here to resources which other scholars call a "lexical knowledge base" (for instance San Vicente et al. (2014)), as they offer many possibilities to leverage the resource. An example is to traverse specific annotated relations (e.g., synonymy, antonymy) and hence model the expansion in a graph-based manner.

The corpus-based approaches on the other hand rely on statistical measures based on different proposed foundations like sentiment consistency (Hatzivassiloglou and McKeown, 1997), point-wise mutual information (Turney, 2002), context coherency (Kanayama and Nasukawa, 2006), double propagation (Qiu et al., 2011) or label propagation (Huang et al., 2014).

As resources such as WordNet (Miller, 1995) are not available in many languages in such comprehensive versions as for English, there were also attempts to leverage machine translation to create such resources for other languages. But this method leads to a non-negligible amount of work for the correction of the output (see Vicente and Saralegi (2016)).

Velikovich et al. (2010) aim at the creation of lexical resources using a large web corpus (4 billion web pages). They use the co-occurrence statistics of n-grams to construct a representation in the vector space from which they subsequently calculate cosine similarities to model a graph. Finally, they use a modification of the label propagation approach to leverage the graph, given a seed set of known polar words to expand the lexicon. This approach makes it obvious that the so-called dictionary-based approach and the corpus-based approach also work in combination.

An approach that takes another route is presented in Severyn and Moschitti (2015). Here, the goal is to use data (tweets in this case) on which a SVM classifier is learned,

using distant supervision[41] and subsequently attribute the features (n-grams) with a sentiment association score given the weights from the learned model. While this leads to an extremely large "lexicon" (in this case 3 million entries), the authors also state clearly that they do not intend to produce human-readable lexicons but rather choose to leverage a corpus and distant supervision to automatically generate the resources.

In recent research there were also many suggestions to use word embeddings (cf. Mikolov et al. (2013b)) for the task of lexicon induction. Similarly to the idea to use document labels via a learned classifier, Tang et al. (2014) included the document-level label into the calculation of embeddings. In a second step, they use the resulting embedding to represent words and learn a classifier based on a small labeled seed set to predict the sentiment score of other words. This idea was further extended concerning the embedding calculation by Wang and Xia (2017). They do not only use document-label information but also word-level information and combine it in the cost function of a joint learning approach in order to create a specialized representation.

Rothe et al. (2016) learn an orthogonal transformation of the embedding space in a manner which leverages sets of words of opposing meaning (positive-negative, concrete-abstract, frequent-infrequent). The resulting embedding is reduced to low dimensionality (in the extreme case to a single dimension; hence the name Densifier) which represent a certain type of information (the opposing meaning). The objectives to learn the transformation are to maximize the distance of opposing words and to minimize the distance of words within a set. For the lexicon induction they apply a reduction to one dimension so that all words in the embedding are represented by only one value, which is a proxy to estimate to which extent words are coupled to this axis of information. In other words, in the resulting embedding, each word is represented by a number, directly being interpretable as a ranking according to the information represented by the given seed sets.

Hamilton et al. (2016) propose SentProp, an algorithm to derive lexicons from an embedding. In their approach, they first create a graph of the embedded words, i.e., they link each word to its k nearest neighbors (according to cosine similarity) and calculate the weight of the edges. In the next step, they use seed words (which bear positivity or negativity) from which they perform a random walk in the graph to propagate the sentiment labels. As a result, the sentiment score of a word is proportional to the probability of the random walk (starting from one of the words in the seed set) to hit the target word.

[41] In their work, as in many other social media content based sentiment analysis systems, the authors use emoticons as indicators for the distant supervision.

An et al. (2018) use semantic axes to score words with respect to these axes. To produce such an axis, they use the vector of a pole word (or the average over a set of pole words) and subtract the opposing pole word (or averaged pole word set). The resulting vector is hence in the direction of the semantic axis defined by the pole words (or word sets). In the next step they just compare an arbitrary word to this axis using cosine similarity and obtain a score which signifies the distance of the word to the subtracted pole vector, or the similarity to the pole from which the other pole was subtracted.

3.2.2 Most Related Lexicon Induction Approaches

Considering the lexicon induction component (described in Chapter 4), there are several levels we can refer to, such as the required resources, the possible applications, or the basic principles of the algorithm used for the implementation.

If we refer to the backing source or the modeling, our approach is different from dictionary-based and corpus-based approaches since it does not rely on linguistic knowledge resources such as WordNet, or on applied statistical measurement given a specific corpus. Instead, it relies on an embedding model[42]. In this sense, we consider SemAxis from An et al. (2018) who also use a word embedding for lexicon induction as the closest to our proposed approach.

As described above, they rely on a set of antonyms to identify semantic axes in the embedding space in order to measure the relative positioning of words against those axes (using cosine similarity). In contrast to this, our proposal *can* incorporate such antonymic information (as is shown in the experiment inducing a sentiment lexicon from Section 6.3), but it does not require two extremal poles of a concept.

While many of the given references are dealing mainly with sentiment lexicons (and hence dealing with negativity and positivity), the applicability for other lexical realms cannot be safely assumed without further investigation[43]. Although many approaches have no fundamental restrictions concerning the application to other concepts than sentiment, many of them focus on the semantic differential (Osgood et al. (1957)) of a lexical dimension and are thus limited to such cases where this semantic differential exists.

[42] It is important to point to the perspective that an embedding is somehow a mixture of a linguistic knowledge base and corpus statistics. Using the distributional hypothesis, a calculated embedding contains information about similarity (and relatedness) in meaning. But this information is not manually annotated and formalized by experts as in WordNet. Rather, we find the different kinds of relations intermixed and have to cope with this. In addition to this, since the embedding is lastly based on statistical information from a corpus, there is also a link to the corpus-based methods. However, with an embedding alone, we are not able to count for example the occurrences of words with certain connectives (like *and* or *but*) and therefore an embedding is clearly a different backing model than a corpus.

[43] It is for example questionable if the required amount of (distantly) supervised material would be available if one cannot rely on large collections of social media messages and leverage a simple indicator like the emoticons for the labeling.

We demonstrate in Section 6.4.1 and 6.4.2 that we are able to use the same approach to derive and extend lexicons from positive (not in the sense of sentiment) examples only. Additionally, we address also other purposes than the induction of sentiment lexicons and emphasize the broad field of applicability. For instance, we also leverage the model's feature of semantic combination to combine concepts (communication and negativity in Section 6.2.1; crime and the financial sector in Section 6.2.2) to steer the induction process in the desired direction.

While An et al. (2018) use a similar scheme to cluster and work with the centroids, they do not use the combinatory aspects of different concepts to derive new resources.

If we look at the level of the algorithm, we consider Gyllensten and Sahlgren (2018) as a related piece of work. Although not introduced as a method for lexicon induction but as a method for term set expansion (applied in the realm of information retrieval), the authors have proposed a usage of embeddings in their work which is similar to ours. While the intended field of application is certainly different (query term expansion vs. lexicon induction), a slight resemblance—especially for the iterative approach—can be noticed. Due to the different targets for application and hence different requirements, our work differs substantively in a wide range of details (see also Chapter 4) and specific decisions for implementation.

3.2.3 Summary

There are many propositions and possibilities to create lexical resources. As the manual creation and extension or adaption is seen as costly and cumbersome, researchers have tried to leverage corpus-based information or more formalized linguistic resources such as WordNet (in the sense of compiled linguistic knowledge base) to extend lexical resources. Furthermore, these approaches are often starting from a small seed set which is then used to propagate this information and hence identify further candidates for the lexicon. Many of these approaches have also used a mixture of corpus-based information and linguistic knowledge to create even better resources.

The intuition behind the idea to leverage the distributional representations of words is also clear, given the general schema to bootstrap a given seed set and using an additional resource which provides opportunities to identify similar candidates. In the case of an embedding as model for distributional semantics, we may be forced to cope with a general notion of similarity and cannot rely on such structured information as it is present in WordNet (consisting of many formalized relations like antonymy, sinonymy, hypernymy, etc.). Nevertheless, scholars have demonstrated more recently how to use embeddings to derive lexical resources.

One possibility is to incorporate supervision information into the calculation of the embedding and then use the specialized representation to create the respective lexicon. Another way is to start off with an already calculated embedding. In a purposeful transformation step the embedding is then adapted to reflect the basic information (the contrasting juxtaposition of two given seed sets) to facilitate the derivation of a lexicon. Alternatively, the information about similarity in the embedding space may also be used in a straightforward way to model a graph and use a seed set which is further propagated through the graph. Finally, the seed set may also be used to directly calculate a vector that represents a semantic axis in relation to which a set of given words are scored.

3.3 Text Classification

While classifying units of text on the document-level is clearly one of the main possible applications of lexical resources, content analysis methods do not adhere to such a restrictive definition. Rather, we often find smaller units of text (or no defined unit of application for the classification at all) which are measured with respect to the phenomenon under investigation. The goal of the content analysis does not necessarily deal with considerations about structure, shape or length of the unit of analysis in the first place. However, since we perform comparative experiments in the empirical part of this work which are exactly about the task of document classification (see Chapter 7) or are in close relation to it (see Chapter 8), we will give a brief overview of different related approaches in this section.

In order to make this section also accessible for readers from the field of social sciences, we will—after referencing survey literature below—include a brief introductory section that explains the task and how it is tackled in a standard fashion. The reader who is already familiar with the task of text classification and the standard methods might indeed just skip this section.

As the field of text classification is vast[44], we refer to the following surveys which contain much more detailed descriptions of the specific methods and techniques which have been proposed for this task.

Aggarwal and Zhai (2012) give a concise overview on the mostly widely used algorithms and also on some of the preprocessing steps. Mirończuk and Protasiewicz (2018) enlarge the scope of their survey in the sense that they also include precedent phases like text acquisition into their description of the task. Additionally, they update the survey from Aggarwal and Zhai (2012) through the inclusion of newer work and analyze the trends in

[44] Actually, a noticeable part of conference papers in computational linguistics use this task for extrinsic evaluation of proposed methods and resources.

this field. Finally, Altınel and Ganiz (2018) focus on semantic text classification which tries to overcome some of the weaknesses of standard approaches that use a Bag-of-Words (BoW) text representation coupled with vector-space classifiers. For more introductory and explanatory sources, the interested reader may also refer to the according chapters in Jurafsky and Martin (2019) and especially in Manning et al. (2010) for the modeling aspect as well Witten et al. (2016) which is rather focused on the algorithmic part.

3.3.1 Text Classification in a Nutshell

The task of text classification is to predict a label (of a class) for a given piece of text. To carry out this prediction automatically, we need an algorithm that assigns the label. In simple cases, the algorithm may consist of a set of rules which are discriminative enough to classify the text[45].

To improve the performance of such a classifier, we want to systematically incorporate knowledge about the domain into the classifier. While manual creation of those rules and incremental improvement is possible, we may also wish to automatically create such an improved classifier.

In the case of learned classifiers, we leverage annotated data to learn a model which predicts a label for new, non-labeled texts. This scenario is called supervised learning[46]. In order to aptly classify new, unseen texts, the model needs to be able to generalize, i.e., the annotated data should be used to automatically derive a model that does not only memorize the presented texts with the according labels but is able to form a more general base for the prediction that abstracts from the single examples.

While many different approaches to learn such a classifier exist, the first step to learn from *textual data* is to transform it into an apt representation. Because text is not a set of quantitative measurements on which we can directly apply the algorithms to learn a classifier, we need to transform the text into such a form. A typical way is to model the text as a Bag-of-Words. In this representation, we split the text into words and further encode the text document into a vector where each dimension of the vector represents a specific word and the value in this dimension is the number of occurrences of the word in the text[47].

[45] For instance, we assign the label "sport" to a news article if we see one of the following terms: *football*, *ice hockey*, or *tennis*, otherwise we do not label it as "sport". Although this rule seems to make sense, it will produce many false negatives (articles not tagged as sport although they are articles about sport) and a number of false positives (articles which contain one of the words, but which are not about sport) as well.

[46] The supervision refers to the annotation of the data, i.e., the labels are the fossilized form of advice

[47] See Erk (2012) for a general introduction into vector space modeling.

While this conversion of textual data into a quantified representation is straightforward, there are also some drawbacks that come with it. First and foremost, all words are modeled equally, i.e., we measure the occurrences of the word *the* in the same way as we measure the word *tennis*—obviously not a very advantageous property of the representation, given that words differ in how much information they bear. According to Zipf's law (Zipf, 1949), the frequency of a word is inversely proportional to the rank of the word in a frequency table. As a consequence, the simplistic version of vector space modeling of textual data is hence dominated by the high counts of a few words which occur in almost every document numerous times but carry little information about content, e.g., determiners like *the* or *a*, conjunctions like *and*, pronouns, and so forth[48]. The most direct strategy to counteract this phenomenon is to ignore such words. This filtering is in fact often applied as a preprocessing step and is called stop-word removal.

While this filtering technique alleviates some of the problems of the vocabulary distribution in text collections—from which we would like to learn specific patterns for the classification task—the importance of specific terms to contribute meaningfully to perform the prediction task is still unclear. In order to focus on the words that contribute the most for the distinction of the textual units into separate classes, one uses often a TF-IDF weighting (Salton and Buckley, 1988), a technique stemming from the realms of information retrieval where it is especially important to focus on the most important words of a document. TF-IDF extends in principle the term frequency counts through a weighting factor which results in a lower weight for words that occur in many documents in the collection.

This is in line with the requirement for a classifier algorithm, i.e., to discern between documents of different classes (labels). Here, it helps to focus less on words that appear in almost every document in the collection which decreases its differentiation potential for the text classification task. Thus, with the step of TF-IDF weighting we also leverage the information about the distribution of terms within the document collection in a data-driven way.

In the next step of a typical text classification approach, one uses this (weighted) Bag-of-Words representation of the textual unit to train a classifier on the given data. Aggarwal and Zhai (2012) point out, that many algorithms have been proposed for text classification, such as Support Vector Machines (Joachims, 1998), Naïve Bayes (McCallum et al.,

[48] Additionally, this kind of representation leads to high-dimensional sparse vectors, i.e., for most words, there are many documents that do not contain those words. Or, to put it differently, a document consist typically only of a small subset of the full vocabulary of the union of all documents that are taken into account. Since the documents are represented by a count per word (and hence a dimension per word), this leads to many counts with value 0—also related to Zipf's law in the sense that most types are rare. Since this sparse high-dimensional representation complicates the inductive learning from such a feature space, this fact is also often referred to as "the curse of dimensionality".

1998), Decision Trees, or neural network classifiers, just to name the most prominent ones. The goal in this step is to train a model that predicts the labels of the given annotated data while preserving the properties of generalization, i.e., we attempt to learn a model that automatically identifies the patterns in the data which allow for the discrimination of texts between the different classes.

While the above mentioned processes for the learning of such a classifier generally do not require any adaptation for specific cases of text classification—and still perform astonishingly well (see for example Wang and Manning (2012)), given their simplistic modeling of textual units—there are many methods to improve the performance which lead into the direction of specialization. This means, that the further steps towards the improvement of the classifier are made in a way which makes the classifier more specifically suited for the case at hand[49].

One possibility is to adapt the feature space, i.e., to adapt what is represented and in which way in the vector space modeling. While TF-IDF as a weighting scheme may also be seen as such an adaptation, another way is to apply feature selection (see also Aggarwal and Zhai (2012, 167ff). Feature selection refers to techniques where we statistically determine which words should be kept in the feature vector, i.e., which words will be taken into account for the classification at all. There are also methods (often referred to as feature extraction) which apply clustering methods or factor analytic methods (like SVD) on the original word-based feature space to transform it into another feature space with the goal of facilitating the training of a classifier on the transformed space.

Another way to improve the classifier is with (manual) feature engineering. This refers to a method where we define in a deductive way—often led by our intuition how we as human beings would solve the problem—additional features (dimensions in the vector which represents the textual unit) which are geared to capture a specific type of information.

A good example is the count of words which are denoted in specific lexicons, e.g., for sentiment analysis, we would additionally count all words that are present in the lexicon with negative terms and separately count all words in the document which are present in the lexicon of positive words. In the simplest modeling, these features are not additional features but in fact the only ones on which a classifier bases its decision (see for example Grimmer and Stewart (2013, p. 274f)). But nothing prohibits the combination of data-driven features and conceptual ones and the axis of feature engineering presents a formidable way to inject prior knowledge into the classification process.

[49]The downside of these improvements is that they are often case-specific and their contribution to an improvement cannot be assumed as guaranteed for other cases and has to be verified empirically.

Many of the approaches to improve a classifier also work well in combination. But it is also important to acknowledge that it often depends on the concrete case at hand which methods (and combination of methods) work best with what parametrization. Often, a linear classifier in combination with a TF-IDF feature weighting results in a strong baseline which is not easy to beat with alternative approaches[50].

3.3.2 Most Related Classification Approaches

We present in Chapter 5 a method to apply the lexical resources to the standard task of document classification (see Chapter 7 for empirical evaluation; and also for framing detection, see Chapter 8) which is based on an embedded modeling of the text and the lexical resources as well.

While there are many ideas that are related to the proposed schema for classification, they often stem from different fields, i.e., our proposed method contains elements which are similar to elements of other approaches but which themselves differ substantially from an overall viewpoint. Furthermore, the proposed method is designed according to the guidelines we have defined in Chapter 2, especially simplicity and interpretability.

Hence, we do not consider the classification approach as specifically innovative *per se*, but rather as a good fit to the given requirements and application opportunities. However, in this section we relate the specific elements to research which, in our view, comes closest to the proposed implementation.

If we focus on the pure algorithmic classification level, the proposed approach is most likely to fall into the category of proximity-based classifiers such as the Rocchio classifier (Rocchio, 1971) or k nearest neighbors (kNN) (see (Manning et al., 2010, chapter 14)), since we compare textual input to centroids[51].

But in contrast to many other approaches, we do not derive the centroids directly from the labeled documents and we also do not compare the whole input text to the centroids. Instead, we use a preceding step to derive a lexicon for each class and re-embed the lexicon into the embedded space to derive the centroids for the class (see Chapter 5 and 6).

[50] Hence, we have also decided to use such an approach as the baseline for our experiments in Chapter 7 and 8

[51] There is also an element of the kNN in the sense that we restrict the nearest neighbors (concerning cosine similarities) to the sentences which we take into account to k nearest neighbors. But since we project each lexicon into several centroids and we apply the nearest neighbors constraint on the number of centroids and not on instances or classes, this is not exactly the same schema as we encounter it in kNN.

Since we use a multitude of centroids to represent the *lexical resource* in contrast to centroids which represent the instances of a document class, our approach differs clearly from a standard proximity-based classification. Furthermore, we do not compare to centroids on the document-level but rather deduce the decision on the document-level from the centroid comparisons on the sentence-level.

When we turn to the representation or feature-level, our approach is similar to the Bag-of-Concept approaches that represent a document in a feature vector which does not count the occurrences of words but the occurrences of concepts (see for example Sahlgren and Cöster (2004)). These approaches strive to overcome some of the problems of the BoW approach, so that they allow us to handle synonymy and to project the information from singular instances (i.e., words in documents) to a more general level (concepts in documents), which in turn should increase the generalization potential of the learned classifier.

The idea to use concepts to represent the content of a text (rather than just the words found in it)—on which we also heavily rely—is also present in other research. Dhillon and Modha (2001) use the term "concept vector" to refer to a centroid resulting from a spherical k-means clustering[52] on vectorized representations of texts. The authors note that besides the useful decomposition of documents with the help of the concept vectors (and hence dimensionality reduction), one of the main interesting properties is revealed when one inspects the centroids in the sense of analyzing their nearest neighbors. In contrast to the singular vectors (retrieved by SVD), the experiment shows that the concept vectors are localized around the underlying concepts (the texts that were used to assemble a collection in the first place) and are easily interpretable for humans.

This notion of interpretability is close to what we propose, although we construct the centroids from a lexicon (which is in turn a collection of terms referring to a concept). Also, we do not infer a general set of concepts but rather relate our concepts to the classes from the text classification task[53]. In other words, we work with only one concept per class that we want to predict, which differs from schemas as presented in Sahlgren and Cöster (2004). In addition, we use a clustering step (see Chapter 5) to

[52]This k-means clustering method uses explicitly the cosine distance (or cosine similarity) instead of the euclidean distance. In normalized embedded spaces—such as the ones from a word embedding as a result of an application of the distributional hypothesis as we use it in our work—however, there is a linear correlation between the euclidean distance and the cosine distance.

[53]Apparently, Zhang et al. (2015) applied a clustering on the word embedding as well to create a 5000 cluster based representation of documents. However, since they did not include any further guidance into the clustering process, the outcome of such a procedure should be taken under scrutiny first. The results from their benchmark are at least indicating that the centroids they calculated in this way are not optimal for the task.

identify sub-concepts of a concept (represented by a lexicon). The result of this clustering procedure (the centroids) is further used as a set of proxies to identify the concept(s) under investigation.

A piece of work that comes closer to our approach in the sense that it also uses an embedding to derive concepts via a clustering step is presented by Kim et al. (2017). In this approach, the authors calculate an embedding on the documents and subsequently cluster it (using spherical k-means) to identify concepts (i.e., words that are—according to their cosine distance in the embedded space—grouped by the clustering algorithm). They use the cluster membership of the words to represent documents as a Bag-of-Concepts and apply a weighting scheme (an adapted version of TF-IDF for the cluster representation) and finally learn a classifier for the prediction.

Kim et al. (2017, p. 343) evaluate "representational effectiveness" (they compare different document representations via a triplet task in which one has to determine which of three given documents are the most similar) where the Bag-of-Concept approaches deliver the best results. However, if the full classification is carried out (i.e., an SVM classifier is trained on the respective representation), the Bag-of-Words representation outperforms the Bag-of-Concepts approach on two out of three data sets, again pointing to the robustness and strength of this baseline.

In our approach, we also use an embedding as the backing representation. But in contrast to the Bag-of-Concepts approaches, we do not derive a generic set of concepts or cluster the embedded space trained on the documents from the classification task.

As our goal is to link to the established dictionary-based methods in social science, we follow the strategy to represent the concepts of interest (the classes of the classification task) as a lexical resource. In Chapter 4, we present a method to largely automate the induction of such resources which is time-consuming when performed manually. After we have derived such lexical resources, we do not apply a standard approach for dictionaries but rather re-embed the lexicon and cluster it. The result of this procedure is a set of concept detectors[54]. These are points in the embedded space which represent sub-concepts of the lexical resource. Subsequently, we perform a *sentence-wise* comparison of the (embedded) input and use the cosine similarity to measure the signal of each concept detector which we then aggregate over the document to produce the prediction.

Clearly, there are similarities to the proximity-based classification approaches, but our centroids (or: concept detectors) are calculated from the lexical resources and not from

[54]Hence we name the approach "ABCD", an acronym for "all based on concept detectors", see Chapter 5 and 7.

labeled documents. Furthermore, the prediction on the sentence-level allows us to perform a much more fine-grained prediction (and interpretation) than document-level modeling. Additionally, since we represent the lexical resources not as one centroid but instead as several centroids, we also get another layer of detail through the division into n clusters.

Another view on the concept detectors is to interpret them as higher-level, crafted features on which one could train a classifier. However, for the sake of simplicity and to evaluate the potential of the approach, we applied our classifier in the empirical part of this work in a purely heuristic way. That is to say, we leveraged the annotated data only to induce a core for lexicon induction and did not use any supervision to learn a classifier for the prediction step.

3.4 Chapter Summary

In this chapter we have referred to related work in different areas and positioned this thesis in relation to other research. Since our approach to the automated induction of lexical resources and the application of those resources for fine-grained classification uses an embedding as the backbone, we have also devoted a section to embeddings (as models for distributional semantics).

In 3.1, we concentrated on a brief introduction of distributional semantics and how embeddings are calculated, focusing on so-called word embeddings (although the embedded units do not necessarily have to be words.) Furthermore, we considered it to be important to include a discussion on the known shortcomings of such models and the proposals that have been made in order to overcome these drawbacks.

As many researchers conducted detailed comparisons and investigated benchmarks of such models, the calculation of embeddings is no longer a "black art". Therefore the explanation for their inner working has been improved, resulting also in the empirical finding that default parametrization is quite robust. Nevertheless, if such models are used as the basic layer for other applications, as we propose in this thesis, the users should be aware of their imperfectness, hence our emphasis on the shortcomings. The good news is that we also observe ongoing progress in research to get a better understanding for those models, resulting in a wealth of ideas to improve them.

Since we use an embedding for a sentence-based modeling in our classification approach, we also refer to work that focuses on the question how such a modeling may be transferred to the sentence-level. While there are many approaches that learn a sentence-level representation in a manner that introduces opaqueness (in the sense that such sentence

embeddings cannot be as easily inspected as word embeddings and are primarily evaluated in extrinsic ways or via probing tasks), we rely on a simplistic sentence-embedding (see Chapter 5). On the one hand, this modeling ignores a good part of the structure of the sentence, but, on the other hand, it shows excellent properties considering the encoding of the content, i.e., the words which are taken into account[55].

In 3.2, we presented different approaches that automatically derive lexical resources. A considerable amount of this work refers to the induction of sentiment lexicons. While this is a prominent area of application for lexical resources, in this work we propose a lexicon induction method that is in principle not limited in any way to a specific domain. Additionally, we emphasize the combinatory opportunities that our embedding-based approach incorporates—a quality that still holds potential to be exploited further.

Finally, in 3.3, we discussed text classification which is a vast field since it is one of the basic applications of language technology. In this light, we refrained from an encompassing overview of the literature (which lies beyond the scope of this work) and pointed to multiple surveys which are in turn a compressed view on this field. Instead, we described in brief the most important steps of standard text classification methods in order to make this part accessible for researchers from the social sciences. Because it is clear that this compromise is not an optimal solution, we also gave some reference for more introductory and detailed literature.

[55] Because this property matches our requirements to apply the lexical resources and it additionally allows us to keep the classification on a high level of transparency, we propose in our work to apply this kind of modeling. Nevertheless, we consider an alternative sentence embedding approach to be a valuable element for possible improvement.

4

Deriving New Lexical Resources Using Word Embeddings

> "The best way to get a good idea is to have a lot of ideas."
> — Linus Pauling

```
In [57]: analogy(a="Lexikon", b="Wort", x="Musik", y=None, model_given=model, verbose=True)

'Lexikon' is to 'Wort' as 'Musik' is to 'Klang'

Out[57]:
[('Klang', 0.5736246705055237),
 ('Ton', 0.5524649620056152),
 ('Gesang', 0.524972677230835),
 ('Melodie', 0.5196679830551147),
 ('singen', 0.48464903235435486),
 ('Lied', 0.481370747089386),
 ('Sprechgesang', 0.4732584059238434),
 ('Song', 0.4717941880226135),
 ('hören', 0.4716670513153076),
 ('Sound', 0.46963757276535034)]
```

In this chapter, we describe an approach to derive lexical resources for arbitrary concepts. More concretely, we strive to assemble lists of terms which are semantically linked to

these concepts. Throughout this work we will refer to these resources as "lexicons" as they are also referenced in the literature for lexicon-based approaches (e.g., Taboada et al. (2011), Hamilton et al. (2016), Hu and Liu (2004))[1]. No additional information neither in the sense of morphological information nor in the sense of a description of the terms is assembled; a lexicon is thus simply a list of terms.

We create this kind of resource to satisfy the given need to link to established methodology in social science for (automated) content analysis of textual input. More precisely, we follow one of the most popular techniques in the field by creating lexical resources (often also called dictionaries) which serve the purpose of identifying a given category or concept (cf. Schwartz and Ungar (2015), p. 81).

As a guidance for the reader, we recommend to first read the introductory examples from Chapter 6 if the reader's interest is geared to get a quick overview on the mechanics of the proposed algorithm. Since this chapter is focused on the detailed description of the algorithm itself, it may serve the reader better to read it afterwards if example-based explanation is the preferred way to learn about a method.

4.1 Deriving a Lexicon from an Embedding

As mentioned in Chapter 3, there are several possibilities to derive new lexical resources. These are further categorized into approaches that use a corpus (and its statistics) and other approaches that use a specific knowledge resource to propel the process of lexical induction. Our approach relies heavily on a model of distributional semantics that incorporates the meaning of words or terms by distributing them in the semantic space (cf. Sahlgren (2006)).

Therefore, this approach has to be placed somewhere in the middle between corpus-based and resource-based lexical induction. On the one hand, the embedding model is comparable to a semantic resource such as WordNet (Miller, 1995) in the sense that semantic similarity (and semantic relatedness) is modeled and quantified (although only in a relative measurement and normally mixing different semantical relations, as mentioned in Chapter 3). On the other hand, the embedding model is derived directly from

[1] In a sense, we could also use the even more unspecific term "word list" as it is used for example in Wilson et al. (2005) for a reference to the lexical resources from the General Inquirer (Stone et al., 1966). However, we think that it is helpful to use the term "lexicon" as in other related work.

The term "word list" is too specific because the unit of the entities in the lexicon is linked logically to the units represented in the embeddings which often also include n-grams identified as collocations or phrases (cf. Mikolov et al. (2013a).) As described in Gyllensten et al. (2019), also longer *and* shorter units could be suitable to improve the quality of the embedding. Also, we use "lexicon" instead of "dictionary" as it is used for example in Grimmer and Stewart (2013).

a large corpus of text without further curation of the pure text data or injection of outer knowledge into this process.[2]

Throughout this thesis, we make use of a `word2vec` word embedding model (Mikolov et al., 2013b) which incorporates a distributional semantics model for the German language. We will furthermore also refer to this model as *semantic space* or simply *embedding(s)*.

The text collection that we use as a corpus to train the model consists of articles from the three biggest newspapers in the German speaking part of Switzerland (Blick, Neue Zürcher Zeitung, Tages-Anzeiger) from the years 2004-2015. This results in a collection of 1.253.479 articles. Since we focus on semantic relatedness, all texts were lemmatized.[3] This leaves us with a corpus of roughly 430 million tokens.

After experimenting with an evaluation set for German embeddings[4] we set the dimensionality of the semantic space to 400, apply a cut-off for a minimum count of 20 after we applied the standard phrase detection implementation[5] from the `word2vec` model in `gensim` (Řehůřek and Sojka, 2010). We use the skip-gram algorithm with negative sampling (10 negative samples) on a window length of 5. This setting results in a vocabulary of 416.501 terms. This model is further used to determine similarity of given terms or for given points in the embedding space.

4.1.1 Intuition by Example

Let us start with a simple example. Since we know that in the embedding similar words (words with similar meanings) are close together (considering cosine similarity), we first have a closer look at the most similar terms for a given term. When we investigate the resulting top 10 nearest terms to "Roger_Federer", we observe that many of the top entries are "of the same kind" (successful tennis players), or, in other words, semantically

[2] One could consider lemmatization as an injection of outer knowledge and—since we used a lemmatized corpus to train the embeddings—therefore argue that the embeddings are not derived from pure text data. However, we would argue that lemmatization is a rather generic process only simplifying the learning of the meaning, and that the representation of the tokens with their base form is a generalization which does not introduce further semantic relations. For the lemmatization we apply the TreeTagger (Schmid, 1994) and use the surface form if no lemma is provided. Although the lemmatization is imperfect, the erroneous lemmatization is largely consistent and therefore not a problem for a downstream application if the same (erroneous) lemmatization is applied in.

[3] Additionally, punctuation was removed. But the word forms were not lowercased and numerals were also kept in the raw text.

[4] https://devmount.github.io/GermanWordEmbeddings/

[5] The underlying scoring function selects bigrams based on a mutual information criterion. The process may be repeated for longer n-grams (see Mikolov et al. (2013b), Mikolov et al. (2018)). However, we apply only one round of phrase detection for our embedding model and accordingly get only n-grams of length two. In practice, this leads often to "phrase tokens" which represent names of actors, like *Roger_Federer*.

closely related concerning this aspect. Consider the list of nearest neighbors in the word embedding model for the given term *Roger_Federer* in Table 4.1:

10 most similar entries to "Roger_Federer"		
Rank	Word	Similarity
1	Federer	0.8865
2	Rafael_Nadal	0.8700
3	Stanislas_Wawrinka	0.8609
4	Novak_Djokovic	0.8475
5	Nadal	0.8329
6	Wawrinka	0.8277
7	Wimbledon	0.8214
8	Djokovic	0.8172
9	Stan_Wawrinka	0.8159
10	Andy_Roddick	0.8091

TABLE 4.1: 10 most similar terms to *Roger_Federer* in the semantic space of the word2vec model, ordered by cosine similarity

We see that we find other tennis players (*Rafael_Nadal*, *Stanislas_Wawrinka*, *Novak_Djokivic*, and *Andy_Roddick*). We also find synonyms for the given term (*Federer*) and for the other tennis players. Additionally, we find *Wimbledon* which hence seems to be—not surprisingly—a term that is also often mentioned in the context of the given term *Roger_Federer*.

If we want to create lexical resources which contain lists of terms that identify a concept (either by being instances of the concept or semantically linked terms), we should be able to leverage the modeled meaning of words in a word embedding.

In the next sections we first explore some more examples and properties of the word embeddings that we use subsequently to implement an algorithm which enables us to create lexical resources for arbitrary concepts.

4.2 LexExpander: An Algorithm to Generate Lexicons Based on Concepts

While we have seen that the similarity of words is intuitively a good starting point for lexical expansion, it is also clear that we need to avoid oversimplifying procedures to create a coherent lexical semantic resource. This is especially important due to the partly unpredictable property of the embedding to also place words that are semantically related (and not semantically similar) close together in the embedding space.

Sahlgren (2006, p. 24) argues:

> [...] that the inability to further qualify the nature of the similarities in the word-space model is a consequence of using the distributional methodology as discovery procedure, and the geometric metaphor of meaning as representational basis. The distributional methodology only discovers differences (or similarities) in meaning, and the geometric metaphor only represents differences (or similarities) in meaning.

This means in turn that if our goal is to work with standard word embedding models, we have to accept this mixture of different semantic relations (synonymy, antonymy, hyponymy, meronymy, and so forth) as all being part of the semantic similarity of the model (cf. Sahlgren (2006, ibd.)).

This is not to say that an embedding could not be calculated with specific relations as intentional guidelines—on the contrary, there have been many methods describing such adaptations by either modifying the objective function of the embedding algorithm (see Section 3.1.3) or re-arranging the elements of the calculated embedding space themselves (e.g., retrofitting them (Faruqui et al., 2016)). But since an a priori restriction on a specific type of relation is not desirable for lexical representation of arbitrary—yet semantically motivated—concepts, we strive to cope with unwanted outcomes through other measures.

4.2.1 Search and Assess

To expand a lexical resource, we have to find new apt candidates which we include in the lexicon. More precisely, this means that in a first step we have to implement a method to *find* the candidates and in a second step *assess* those found candidates to decide if they should be integrated into our lexicon.

For the generation of new candidates for a given lexicon (or a small seed set of terms which form the core of the concept) we will focus on two ways to use the embeddings. First, for the search of new candidates and the guided exploration of the embedding space, we will make heavy use of the recombination of terms. In other words, we use the property of the representation that the meaning of two (or more) entries can be combined via vector addition. Second, for the assessment we will use the nearest neighbors of the new terms to estimate the fit of the candidates for the lexicon.

For the first point mentioned, consider the example given in Table 4.2. We see that the most similar word to *gut* (*good*) is *schlecht* (*bad*). If we wanted to create sentiment

lexicons, this outcome is surely not desired[6]. Furthermore, other particles (*auch, doch, mehr, also, aber, hier*) which do not contain the desired semantic properties are parts of the list[7]. As a result, the query for the nearest neighbors in the embedding space yields a mixed set of words which contains apt candidates (*hervorragend* (*outstanding*), *ausgezeichnet* (*superb*), and *exzellent* (*excellent*)) but also many unwanted terms. A way to counteract this tendency is to combine the original term with a second one to steer the similarity search in such a manner that the result contains terms which are similar to *both* given words.

10 most similar entries to "gut"		
Rank	Word	Similarity
1	**schlecht**	0.7660
2	hervorragend	0.6337
3	auch	0.5464
4	doch	0.5402
5	ausgezeichnet	0.5359
6	exzellent	0.5355
7	mehr	0.5307
8	also	0.5302
9	aber	0.5269
10	hier	0.5265

TABLE 4.2: 10 most similar terms to *gut* in the semantic space of the word2vec model, ordered by cosine similarity

In Table 4.3 we observe intuitively that vector addition of two already closely related terms that both contain the desired semantic property helps to guide the search for new candidates into a desired direction [8].

10 most similar entries to "gut"+"hervorragend"		
Rank	Word	Similarity
1	ausgezeichnet	0.7745
2	exzellent	0.7495
3	**schlecht**	0.7349
4	vorzüglich	0.6584
5	bestens	0.6408
6	perfekt	0.6283
7	erstklassig	0.6192
8	grossartig	0.6102
9	optimal	0.5837
10	toll	0.5825

TABLE 4.3: 10 most similar terms to a combination of the vectors for *gut* and *hervorragend* in the semantic space of the word2vec model, ordered by cosine similarity

While the simple combination of vectors provides an appropriate level of control for the generation of candidates, we still see the antonym *schlecht* (in bold) in the third position

[6] If we were to assemble a general lexical resource comprising of evaluative adjectives, let us suppose for detection of subjective statements, this search result would be totally acceptable.

[7] This is because *gut* is also often combined in syntagmatic relation with them.

[8] Note that—as a trivial consequence of the commutative property of summation—the order of the summands does not contribute to the result. In consequence, a search for most similar terms to *hervorragend* steered by the addition of *gut* will yield the same result.

with respect to similarity. Hence, the combination with a similar term (from the lexicon or as an imposed steering) does not prevent unwanted terms from appearing in the top results of the search, as we see with the antonym *schlecht*, which still is close to *gut + hervorragend*. Since neither the assessment of the cosine similarity *per se* (as a scalar) nor the rank of the terms in the list is reliable in order to predict the aptness of a new term, we need other measures to better estimate the fit of a candidate.

One way to do this becomes intuitively clear when we take a look at the most similar terms of the supposed candidate *schlecht*, as we do in the following.

10 most similar entries to "schlecht"		
Rank	Word	Similarity
1	gut	0.7660
2	miserabel	0.6844
3	mies	0.6146
4	schwach	0.5878
5	hervorragend	0.5623
6	ungenügend	0.5437
7	katastrophal	0.5425
8	lausig	0.5016
9	mittelmässig	0.4978
10	ausgezeichnet	0.4916

TABLE 4.4: 10 most similar terms to *schlecht* in the semantic space of the word2vec model, ordered by cosine similarity

In Table 4.4 we observe that, although *gut* is the closest neighbor and also *hervorragend* (rank 5) as well as *ausgezeichnet* (rank 10) are of the opposite meaning (considering the positive/negative aspect), the other seven terms in the top ten list are rather synonyms to *schlecht* considering negative aspects. Therefore, they are usable as an indirect predictor for the assessment of *schlecht*. In other words: if we check the most similar words to a given candidate, it allows us to estimate how close this word is if we carry out a comprehensive comparison to a given lexicon. In the given case, the assessment should then be based on the fact that 70% of the resulting word list are not in the supposed basic positive terms list.

To summarize the insights gained from the given examples:

- We observe that we find good candidates for the expansion if we use a word embedding model and focus on the nearest neighbors of given terms from the core lexicon
- We note that besides the good candidates we also find antonyms and other words in those lists of nearest neighbors which are not good candidates

- We also note that we cannot use a simple threshold on the similarity value (cosine similarity) and/or the rank since this is not reliably filtering out unwanted candidates
- If we apply vector addition on two words from the lexicon, we notice that this "steering" gives us a certain amount of control over the candidate generation
- Even if we narrow the focus in that way, strong similarities (e.g., antonyms) persist
- If we use an indirect assessment in the sense that we compare the nearest neighbors for each candidate with the given core lexicon, we should be able to distinguish good and bad candidates (antonyms in this case)

In the next section, we will focus on the ingredients needed to perform the lexicon induction.

4.2.2 Ingredients

To create a lexicon one needs to have a proper idea of what concept should be covered by this lexicon. More precisely, one should be able to decide for new terms if they should be integrated into the lexicon (for the sake of simplicity we leave common second-level features of a lexicon such as weighting factors aside).

Our premise here is that at least a subset of the terms which should be in the lexicon (or which are at least closely related to the concept of interest) are known.[9] We furthermore refer to this initial minimal collection of terms as "seed lexicon" or "core of the concept". The main purpose of this core is to serve as a point of reference during the assessment stages of the lexicon induction process.

The approach that we propose uses a word embedding model. Note that the approach is agnostic to the kind of embedding model as long as it does provide a similarity measurement and allows for meaningful recombination of the entries using vector addition. However, we will use a SGNS (skip-gram, negative sampling) model computed with the *word2vec* algorithm (Mikolov et al., 2013b), using the implementation from gensim (Řehůřek and Sojka, 2010).

Since one of our aims is to make the expansion of the core adaptive (either to a direction found by any kind of data-driven process or by an intentional guiding), the algorithm's starting point must be provided. In the most trivial case, this is just a word from the given seed lexicon. However, any kind of combination of terms already included in

[9] Of course, any other kind of data-driven method to derive such a core (see Chapter 6) is feasible—or any other kind of external resource such as WordNet (Miller, 1995) to create such a seed lexicon.

the lexicon as well as non-existing terms is feasible as long as they are present in the vocabulary of the embedding.[10] This combination of given terms is the starting point for the search.

4.2.3 The Shadow Lexicon

When we put the explored search space (the created candidates and the change in the lexicons during the runs which will be described in the next section) under scrutiny, we observe that we often encounter words that are close to the (current) concept but for which we do not have enough evidence to include them into the lexicon. If we just discard those candidates during the process of candidate generation, this might be the best decision for candidates showing up only once. But consider the case when we see a candidate repeatedly occurring during different runs and with different starting points. Here, we should be able to receive and record the weaker but repeated signal to profit from this information later.

To implement such a device, we introduce a second lexicon which comprises of all terms for which we have reduced but still noticeable evidence that they are connected to the current concept. In other words, this container stores (temporarily as we will see) indications for terms which are insufficient to predict if the terms should belong to the lexicon.

We name it *shadow lexicon* because for the terms in the shadow lexicon, we cannot see clearly enough to assess them properly. Spoken figuratively, we would need more light to better estimate them—they are still in the shadows. The term should also emphasize that there is of course the possibility that those candidates turn out rather to be excluded —if more light is available for a proper estimation.

Hence, the shadow lexicon is the container to capture these terms for which we do not (yet) have a sufficient indication to include them into the lexicon. But instead of discarding this kind of information, we keep it in a separate container which is then also included in the assessment process. More precisely, during the assessment step — which relies heavily on the examination of the surroundings of the candidates through comparison with already known content of the lexicon— we also refer to the shadow lexicon and let it contribute with an optional weighting factor to the overall calculation. For more formal details, refer alternatively also to the listing of Algorithm 2 in the next section.

[10] Even this constraint is not valid if the embedding model allows for ad-hoc creation for vector representations, for example `fastText` (Joulin et al., 2016).

Additionally, the shadow lexicon is also a suitable place for words that are related to the concept that we want to cover with the lexicon, but which are not to be included themselves (because they do not match the concept as such, or are too general).

4.2.4 Algorithm

In this section, the algorithm is described in textual form as well as in pseudo code. It may lie in the eye of the beholder which version is more accessible to the reader. For the sake of simplicity the algorithm is split up into two parts: search and assessment.

4.2.4.1 Search

In the first step we create some candidates which we will assess in the second step. To use the aforementioned properties of the embedding model—similar terms are close in the semantic space—we rely mainly on extensive comparisons for the n nearest neighbors of a given term in the embedding space.

Step by step, the search procedure looks as follows:

1. With the given starting point S (one or several given words) we are looking for the n most similar terms in the semantic space of the embedding model E. If one of the given words is not in the vocabulary of the model, it will be ignored.[11] If no word from starting point set S exists in the vocabulary of E, a random sample from the given lexicon L is drawn and used instead.

2. We check for the n nearest neighbors of S (with a default value of $n = 50$) if they are themselves already in the lexicon. Additionally, we first filter out words that should not be included neither in the guiding of the next search step nor in the assessment procedure. This is equivalent to being a member of the exclusion lexicon[12]. The state of terms as being new candidates or already belonging to the lexicon is stored.

3. To prepare the next search step, we create a new starting point, i.e., a new set of terms which we use to search for the nearest neighbors. We implement this by re-sampling from the two lists described in point 2. But first, we need to have a

[11] If the used embedding offers a fallback mechanism for unknown words, like the computation based on subword vectors as it is implemented in `fastText` (Joulin et al., 2016), vectors are computed ad-hoc and hence there are (almost) no unknown terms.

[12] Note: there are no restrictions on the content of the exclusion lexicon, i.e., even terms from the lexicon which should be skipped may also be members of the exclusion lexicon.

look at the interplay of the involved terms: There are two main forces which exert some influence on the continuation of the search.

First, the search is bound more or less closely to the pre-existing lexicon (the seed) and its semantics. More precisely, if we add more terms to the next starting point set from this lexicon (which were found in the result set of the prior search), we will get a result that is much closer to the terms that are already in the lexicon.

Second, the steering of the search is more "daring" if we include more unknown terms (which were found in the prior search but not in the lexicon) for the creation of the new starting point set. The more terms we include from the unknown ones, the bigger the chance to detect other new terms but also the bigger the risk of creating bad candidates. Hence, bad candidates may result if the search is not suitably constrained and progresses too far away in the semantic space from the given concept or lexicon, respectively.

In addition, we may determine for both groups how close the search should be to the previous result point or how far apart the next search point will possibly be placed. This is achieved by making the list from which we sample the respective terms longer or shorter. Shorter means here —because the nearest neighbors are ordered by similarity— that the selection provides a closer binding to the previous result point. In other words, we select k terms out of the top o nearest terms which are in the lexicon and l terms out of the top p terms which are unknown (with $k = 1$ and $o = 4$ as default as well as $l = 1$ and $p = 2$). For example, we create a new starting point set by adding one out of the first four known terms (randomly drawn) and one of the two first unknown (new) terms (also randomly drawn) from the list of the most similar terms for the prior starting point.

4. This procedure is repeated for i iterations (with a default of $i = 10$).

5. When the search part of the algorithm has finished, we end up with a list of new candidates. This list contains actually lists of new candidates (a list for each iteration). Many of the candidates will occur several times, created in the search process in several iterations. This is not surprising, given that we use a mixture of known and new terms to keep the search path under control. Also, with default values, we keep the "step size" during the search small, since we re-sample based on the top four (o) and top two (p) terms . Theoretically, we allow the search to wander further apart from the concept in the semantic space with each iteration, therefore inducing more insecurity for candidates from later iterations. But for the sake of simplicity, we will not use the information of the order within the lists of candidates and between them. In other words, we just treat all candidates equally no matter the order in which they were found.

Instead, by applying the strategy of taking small steps multiple times, which results sometimes even in the re-creation of the exact same candidates, we build a first layer of robustness using the re-occurrence for highly probable candidates during search: While we ignore the order of the candidates, we will still use their frequency (in the list of the lists of candidates) to allow for a filter criterion: assessing only the top z candidates (with default value $z = 100$), given the order by frequency[13]

6. For the filtered list of candidates, we apply the assessment process to decide if the terms shall be integrated into the lexicon, into the shadow lexicon, or be discarded.

Listing Algorithm 1 gives a more compressed overview using pseudo code which is equivalent to the respective verbal step-by-step description.

Algorithm 1
Searching in the Semantic Space for Candidates

Input: Embedding E, lexicon L; optional: terms for starting point S, exclusion lexicon C
Output: List of candidates

```
M ← most_similar(S, E)                          ▷ getting most similar terms as candidates
i ← 1
while i ≤ n do                                                          ▷ n iterations
    list_of_in_lex ← {}                                          ▷ used for re-sampling
    list_of_not_in_lex ← {}                                        ▷ partial result
    for candidate ∈ M do                                  ▷ iteration over candidates
        if candidate ∈ C then
            skip
        else if candidate ∈ L then
            list_of_in_lex ⁺← candidate
        else
            list_of_not_in_lex ⁺← candidate
    list_of_candidates ⁺← list_of_not_in_lex                ▷ update candidates list
    new_S ← resample(list_of_in_lex, list_of_not_in_lex)    ▷ get update for search
    M ← most_similar(new_S, E)
list_of_candidates ← aggregate(list_of_candidates)                ▷ aggregate and filter
return list_of_candidates
```

4.2.4.2 Assessment

In the second step we assess the list of candidates we have created through the iterative search described beforehand. In general, we assume that the candidates which occurred

[13] Alternatively, one could also rely on ratios on this point, meaning that a term should be present in a given ratio of the iterations. However, assessing only the top r terms may be a more pragmatic approach to allow for a filtering for the most probable candidates without further assumptions.

most often have the highest probability to be good candidates. However, the order will nevertheless be ignored in the assessment step and all terms in the candidate list will be treated equally.[14] This means in turn that we are able to simplify the description of the assessment process to a single given candidate as the order and length of the candidate list do not interfere with the assessment algorithm itself.

As we have seen in the introductory part of this section (see 4.2 and 4.2.1), there is no clear indication from the similarity between the terms as such that could serve as a grounding for the decision about the inclusion into the lexical resource at hand (remember that the most similar term to *schlecht* was *gut*). On the other hand, we have seen that the most similar terms to the candidates themselves could provide helpful information for the assessment. Hence, we try to infer the aptness of a candidate for inclusion by inspecting its surroundings in the semantic space, i.e., its nearest neighbors which serve as a proxy of meaning.

Step by step, the assessment procedure for a given term looks as follows:

1. For the candidate c we query the embedding model E for the m most similar terms (with a default value of $m = 30$), resulting in a list MS_c.

2. For each of these terms in MS_c we test if it is present in a) the lexicon, b) in the shadow lexicon, or c) not present at all. If the term is present in the lexicon, this is a piece of evidence to include the candidate. Similarly, this also holds true when it is found in the shadow lexicon; but the evidence is weaker in this case and therefore its contribution to the overall assessment should be weighted accordingly. If the term is not present in both lexicons we interpret this as a piece of evidence that the candidate is not apt.

   To incorporate these three different pieces of information, we apply a triage for the terms and build two scores, one representing the evidence for inclusion ($score_{lex}$), the other representing the counterevidence how unknown the term is in comparison with the given lexicon ($score_{not\_lex}$).

   We use the rank in the list of the m most similar terms to additionally fade out the influence of less similar terms. This is based on the premise, that the most similar terms are closer in meaning to the candidate than less similar terms and therefore are a better proxy for its assessment—or, even more simple: that the order (based on the similarity measure) is a reasonable scale for informativeness of the terms and hence should be taken into account. A simple way is to give a

[14] There is a subtlety in this process, which we should mention: for each candidate we assess its aptness for lexicon inclusion. If we deem a term as being a valid expansion of the lexicon, we add it to the lexicon (although kept apart programmatically). This exerts an influence on the forthcoming assessment in the sense that the newly added terms trigger the inclusion of the next candidates and so on.

weight inverse to the rank. For example, the most similar term gets a weight 30 (with $m = 30$ as default) while the 30st most similar term gets a weight 1.

The score for each term is calculated by inclusion of the rank score and a fixed weight for the specific lexicon (default $w = 1$ for the lexicon, $w = 0.5$ for the shadow lexicon) in which it is found. So the $score_{lex}$ is calculated by $\sum_{i=0}^{m-1} w \cdot (m - i)$ for all terms found in one of the two lexicons. (While separating the score from the shadow lexicon list could be useful for a different summation procedure, we simply combine it here with the score from the lexicon list, i.e. $score_{lex} = score_{lexicon} + score_{shadow\_lexicon}$.) Consequently, for all terms not found in the lexicon or in the shadow lexicon, the $score_{not\_lex}$ is also calculated by $\sum_{i=0}^{m-1} w \cdot (m - i)$ with default default value for $w = 1$.[15]

3. Next, we apply a first threshold based on the aforementioned sums. If $\frac{score_{lex}}{score_{not\_lex}}$ is bigger than threshold t (with a default value $t = 0.1$) the term will be kept. Otherwise the candidate is discarded.

4. Finally, if $score_{lex}$ is bigger than $score_{not\_lex}$ the candidate will be included in the lexicon, otherwise it will be added to the shadow lexicon. Thus, the term influences all further assessments and partially subsequent searches.

   Note that both thresholds take also the $score_{not\_lex}$ into account (in the ratio of the sums and in the simple value comparison). Since $score_{lex} + score_{not\_lex}$ is not constant across terms (due to the different weight factors for the lexicon and the shadow lexicon) these thresholds cannot be an absolute value applied to $score_{lex}$ directly.

Like for the search part, Listing Algorithm 2 gives a more compressed overview using pseudo code which is equivalent to the respective verbal step-by-step description.

4.2.4.3 Combination of Search and Assessment

One of the most important insights concerning this algorithm is that it relies on extensive references and comparisons to the given lexicon, for the search as well as for the assessment. This means in turn that the quality of the results for both search and assessment is dependent on the given resources. Since the algorithm was developed to cope with scenarios where the initial lexical core is very small, it will also work for those

[15] The weights for w can be altered to change the relation of the influence from the different kinds of information; for example, when setting w higher for the shadow lexicon, inclusion will rather be forced (treating the shadow lexicon as an equal or better source for reference than the normal lexicon). When setting w for the unknown words higher, the assessment will be more defensive, penalizing candidates with unknown terms in their surrounding.

Algorithm 2
Assessment of the Candidates

Input: Candidate, Embedding E, lexicon L; optional: existing shadow lexicon W
Output: Update action

$MS_c \leftarrow$ most_similar(c, E, m) ▷ getting m most similar terms for candidate c
for $sim\_term_c \in MS_c$ **do** ▷ iteration over the candidate's surrounding
  $rank\_score_{sim\_term_c} \leftarrow (m - (rank_{sim\_term_c}) + 1)$
  if $sim\_term_c \in L$ **then**
    $score_{lex} = score_{lex} + w_{lex} * rank\_score_{sim\_term_c}$
  else if $sim\_term_c \in W$ **then**
    $score_{lex} = score_{lex} + w_{shadow\_lexicon} * rank\_score_{sim\_term_c}$
  else
    $score_{not\_lex} = score_{not\_lex} + w_{no\_lexicon} * rank\_score_{sim\_term_c}$
if $\frac{score_{lex}}{score_{not\_lex}} > threshold$ **then**
  if $score_{lex} > score_{not\_lex}$ **then**
    $update\_action \leftarrow$ include_in_lexicon
  else
    $update\_action \leftarrow$ include_in_shadow_lexicon
else
  $update\_action \leftarrow$ discard_candidate
return $update\_action$

cases. But what holds true for the quality of the result for the whole algorithm is also true for its individual components. Thus, during search and assessment, every update of the lexical resource fosters the quality of the following assessment and the following searches (given that we do not add candidates erroneously).

It is also crucial to emphasize the importance of the shadow lexicon. This temporary ad hoc resource is created during run time. As a consequence, its contribution to the assessment process is increasing with each run performed while the resource grows. However, it must be noted that the biggest benefit comes from the coupling of this resource with the search process. While the shadow lexicon is being filled with promising but yet shaky candidates from the search process, it is exactly the aggregating mass of "grey" information that influences the assessment so that it has more and more points of reference for every run to better decide about the inclusion of new terms.

Given that we observe an improvement of the resources (the lexicons), and consequently of the performed actions during the iterations, it is naturally the next step to chain several runs of the algorithm. As we have mentioned, the shadow lexicon is an important ad hoc resource. It is therefore also desirable to preserve this resource over several runs, although the contribution is the highest if the shadow lexicon is applied to the assessment for a search which highly influenced the creation of the shadow lexicon itself. In other

words, when we keep the initial starting point for the search constant, the growing shadow lexicon contributes the most to good assessment decisions if kept for several runs.

Empirically, it has turned out to be fruitful to perform at least three runs of the whole algorithm. In this manner, the contribution from the shadow lexicon to select new apt candidates for the lexicon becomes noticeable. But also even more runs with the same search (i.e., the same starting point) tend to return good results. While increasing the number of chained runs will normally increase the result set, changing or adapting the search as the most trivial strategy to create intentional extensions to the resource at hand is often more promising. This is one of the situations when the induction benefits the most from human interaction. Since the simple change of the starting point (i.e., giving a new combination of terms which define what and where we search) often results in productive subsequent runs (see also Section 6.2.2). Hence, the algorithm should be used with several different initializations to get the best results.

The algorithm works iteratively and updates the resources at run time. Additionally, there is also an element of (constrained) random in the search process which is productive but this element also introduces a level of fluctuation in the results. While there are several ways to handle this fluctuation (for example combining or merging the results of different (chained) runs with the same initializations), there is also an interesting element of surprise emerging from these facts. However, it is up to the user to decide how much fluctuation (and surprise) is desired for the task at hand and to take measurements to reach the goal.

4.2.5 Parameter Discussion

While we have described in detail both parts of the algorithm as a procedure, we additionally briefly discuss in this section the parameters used in the algorithm to clarify their influence and to point to a valid range of values wherever possible. It must be mentioned that the given default values are the result of extensive experimenting and application to a set of scenarios. While the defaults should allow for a robust off-the-shelf application, the lack of a theoretical justification remains.

4.2.5.1 Starting Point

This is not a parameter in the sense of a numerical setting for a computation. The combination of terms used for the initial search in the semantic space is important. As an advice, it is good practice to use the query (i.e., the combined terms) on the embedding

beforehand to check via an inspection of the nearest neighbors for plausibility of this entry point. It should be pointed out that this entry point often does not have any overlap with the lexicon, i.e., the resulting candidate set has no matching entry in the lexicon. In this case, the algorithm combines (new) candidates from this entry point and a given number of randomly sampled terms from the lexicon in the subsequent iteration. This mechanism guarantees that the starting point does not need to have an overlap with the lexicon, hence allowing for creative re-combination with other concepts (see also Chapter 6).

The default for the starting point (i.e., if no terms are provided) is to sample randomly two candidates from the given lexicon.

4.2.5.2 Number of Most Similar Terms to the Starting Point

This parameter (referenced with n beforehand) has a default value of 50. It defines the scope of the candidate generation during search. While 50 may appear to be a large range (given the list of most similar terms)—meaning that in this list there is already an increased chance for noisy candidates—it has proven to be fruitful not to narrow down the candidate generation and rather be cautious in the assessment step. Additionally, noise or false positives are also observed in the top nearest neighbors, especially given ambiguous terms. Therefore, narrowing the scope of exploration in this stage is not beneficial.

4.2.5.3 Parameters for Re-Sampling

The parameters for the re-sampling process for the next starting point are the main instruments to constrain the search path. The re-sampling is guided by two pairs of parameters:

- On the one hand, out of the list of known terms (i.e., those terms already listed in the lexicon) we draw k terms out of the top o nearest terms. If we set $k = o$, we just get the top k terms, which is equivalent to eliminating the random element from this step. This may be desired to restrict the search more closely to the starting point.
- On the other hand, l terms out of the top p terms from the list of unknown terms are the second ingredient for the recombination. Analogously, setting $l = p$ will remove the influence of randomness from this component.

While enlarging o and especially p will enforce the algorithm to "take a bigger step" away from the last starting point, it also needs to be mentioned that the number of chosen terms (k and l) also affects the predetermination of the next resulting point, given that the query is based on the recombination. In other words, a recombination of only two terms will create a wider search space than searching for nearest neighbors of the mean of ten terms that are already close to each other in the semantic space.

Hence, we recommend to set $k = 1$, $o = 4$ as default, as well as $l = 1$ and $p = 2$, so that the new search is constructed from two terms (one known, one unknown), including a slight influence of randomness to iterate more over the lexicon than on unknown candidates. If a search should be geared to find candidates using the initial starting point, one would increase l. On the contrary, if the search should stay close to the already existing lexical resource, one likely increases k.

4.2.5.4 Number of Iterations

The parameter (which we named i above) is set to 10 by default. It mainly influences "how long" we follow the given search path. Since this path is partly also influenced by random choice (see above) and the dynamically changing resources, the results (this means the entirety of all iterations of one run) possibly differ substantially, although they are based on a common starting point. However, in combination with the other default values, 10 iterations should provide sufficient redundancy for the first layer of robustness. This means that we find several times the same candidates in the surrounding of the n most similar terms of the starting points which are rather closely bound to the given lexicon.

If i is increased substantially, the search will become either more exploratory (given enough random influence) or will turn out to be repetitive and non-productive (given rigid random control), finding the same candidates over and over again. While the first scenario potentially adds more productivity, it will also increase the risk to diverge. The second scenario is not harmful but restricts the productivity. In general, we recommend to use more chained runs (with new starting points and/or preserved shadow lexicons) than increasing the number of iterations.

4.2.5.5 Result Size

The result size z is a filter to restrict the assessment to the (presumably) best z candidates. This affects, of course, also the performance in terms of time consumed for each run, given that the assessment procedure is linear to the length of the candidates list.

But with the caching mechanisms of the implementation and since the default values for the assessment are set rather conservatively, z may be set to 100 or even higher, checking rather all candidates than only the most frequent ones.

However, if we already have a lexical resource with a high coverage (which figuratively has only a few "gaps to fill"), we may set the result size z lower in order to prevent false positives, as the assessment procedure tends to be less constrained if large lexical resources serve as a reference point. In other words, since the candidates are estimated during the assessment based on their nearest neighbors, also terms which are only close to a small part of the (large) lexicon typically show many close connections. Of course, changing the weight of the lexicon to constrain the assessment more harshly also counteracts this phenomenon.

4.2.5.6 Number of Recurrent Runs

As already mentioned while discussing the iterations i, the number of recurrent runs is one of the main factors affecting the size of the expansion. This is no surprise, given that the chaining of several runs (including the optional preservation of the shadow lexicon) increases the probability to iteratively aggregate enough mutual evidence for the candidates (created with the identical starting point for each run) so that the criteria for inclusion are finally met.

As increasing the number of recurrent runs with unchanged starting points is equivalent to forcing the algorithm to find most probable candidates given this initial point, we do not recommend to increase r above 10 without implementing further restrictions.[16] On the other hand, using several slightly different starting points for three recurrent runs (and preserving the shadow lexicon across the multiple starts) has empirically shown good results.

4.2.5.7 Number of Most Similar Terms to Candidate

For the assessment of the candidates we restrict the considered list of nearest neighbors to m terms, with a default value of 30. This value is lower in comparison to n since in this step, we must restrict the criterion of semantical similarity in order to compare with the most reliable references.

During the search, we set n higher, since additional good candidates at the cost of (later filtered out) noise are acceptable. During assessment, it is crucial to prevent too

[16] Another way to prevent the algorithm from accepting false candidates would be to perform not a fixed number of recurrent runs but rather entering a while-loop with an apt stopping criterion, e.g., a number of a priori known bad candidates.

much noise since the newly included terms affect the subsequent searches and even more heavily the subsequent assessments.

Additionally, since we use a rank score based on the number of similar terms to the candidates, increasing m would also affect to ratio of rank scores between the most and the least similar term in the ordered list which is now implicitly given as 30:1. While this could be intentionally decoupled (in the sense of cropping or scaling), we have not further investigated different value ranges, given the other defaults.

4.2.5.8 Lexicon Weights

One of the most important parameters to guide the lexicon expansion is the weight with which the lexicons contribute to the assessment. As a default, we set $w_{lexicon} = 1$ for the standard lexicon and $w_{shadow\_lexicon} = 0.5$. This parameter is, in our experience, the most reliable point to relax restrictions for inclusion of new candidates. If unsatisfyingly few candidates are assessed positively, increasing $w_{shadow\_lexicon}$ to 1.0 or even to 2.0 will lower the barrier substantially.[17] This situation often occurs if we have a small given lexical core (only a hand full of terms) which is additionally not very close to the provided starting point.

Naturally, with a small lexical core as reference for the assessment, the assessment threshold for inclusion is seldomly met (even the weaker threshold for the shadow lexicon). To counteract this situation, we may raise $w_{lexicon}$ to reward especially the cases where we have nevertheless matches with the small given lexicon. But it tends to be more fruitful to increase the $w_{shadow\_lexicon}$, so as to increase confidence in the uncertain information. We recommend this action because it has a self-reinforcing effect: the lowered barrier to include terms into the shadow lexicon fosters its growth. This in turn increases the chance that the nearest neighbors from candidates will be assessed more positively, given the higher sum of weights, resulting from the increased number of terms in the shadow lexicon.

4.2.5.9 Rank Score

As described above, during the assessment step, we use the order of the most similar terms to the new candidates (and not the similarity value as such) as a scaling factor. More precisely, we build the scores which we use for the assessment for the unknown terms and the terms being present in one of the lexicons according to $\sum_{i=0}^{m-1} w \cdot (m - i)$.

[17] We highly recommend the interactive use of the implementation, turning on the verbose mode. The user may observe the behaviour of the search and assessment processes and fine-tune according to the desired outcome.

For the other given thresholds, this results in a higher weight of factor m (=30) for the most similar term to the last one considered (rank 30). In this fashion, we include the confidence in the similarity according to the rank.

While we have experimented with different scaling methods and also with combinations of the rank and the cosine similarity value, we have found that this simple inversion of rank and weight successfully implements the information of the ranked list. However, for other embedding models, a different scaling could turn out beneficial and hence a replacement of the default scaling is rather a matter of the given setting than one on which we could address with a general advice[18].

4.2.5.10 Assessment Threshold

The comparison to the threshold t (with a default value $t = 0.1$) is the main criterion for inclusion into one of the two lexicons. While this is a pivotal point in the whole procedure and affects all other parts of the algorithm, the exact value does not matter that much since its distinctiveness is directly dependent on other factors such as the scaling of the rank score, the weights of the lexicons, to name only the most influential ones.

The value of 0.1 has empirically shown to be useful in experiments for several different use cases. Although the mere implementation of this additional criterion could be deemed as overly cautious, we consider this as another layer of robustness for the process in order to avoid inclusion of disturbing false positives. As we have argued before, we would rather tend to increase the weight (especially for the shadow lexicon) in order to foster the inclusion of more terms if desired.

4.3 General Remarks

As mentioned, the implementation is designed so that both parts of it—search and assessment—are independent *per se*. But it is important to remember that they share the same resources, i.e., the embedding model, and the given lexicons. This is also the link which connects the steps: the search part creates candidates considering these resources and the assessment part decides based on those resources. As the lexical resource gets updated accordingly, this will in turn influence the next run with the

[18] Let us suppose the case, that the top x nearest neighbors of each word have been retrofitted (cf. Faruqui et al. (2016)) to improve the general embedding for a specific use case. In such a scenario, we would use a scaling which boosts the influence of the improved range of words in order to profit on the adaptation of the embedding.

exact same search parameters. Those mutual dependencies are also found between the parameters of the algorithm.

We have given a detailed description of the parameters in order to provide some insights why they have been implemented as levers and switches. But please note that while they may be altered, it still remains a question of the scenario at hand (and mainly the given embedding) how the search and the assessment should be parametrized. To sum up, we have at least the following possibilities to influence the algorithm:

- We may bind the search more closely to the given lexicon or we may relax this restriction in order to include more new candidates—to search further away, so to speak.
- We may opt for longer search paths (using more iterations) before the assessment to explore more space (or in combination with a rigid re-sampling setting create more evidence considering the frequencies)
- Restricting the result size is mainly impacting the performance in terms of speed (focusing only on most promising candidates)
- Lowering the assessment threshold will populate more rapidly the shadow lexicon (at the cost of noise)
- Increasing the weight of the shadow lexicon will force the inclusion of new terms for which there is limited connection to the terms in the lexicon but reasonable evidence from the weaker source of information
- Increasing the number of chained runs (or starting subsequent runs while keeping the shadow lexicon) is productive because with the growing size of the shadow lexicon also its influence on the inclusion decision is increased.

As the main intended use is interactive, the focus should not be on the optimal set of parameter values but more on the possibility to apply and inject as much a priori knowledge of the user as possible to steer and guide the algorithm. Hence, changing the starting point for a new search (for example by picking and choosing terms that are found by the search but not assessed positively—because there is not enough evidence for this aspect of the concept in the lexicon) often turns out to be more productive than tweaking parameters on the same search over and over. And even for such cases there will not be an "optimum" (measured by whatever scoring function) of the parameter settings due to the dynamic change of the backing resources (especially the lexicons), and the random in the re-sampling process.

However, the default values and the respective calculation are chosen so that they should work reasonably for most applications out-of-the-box.

Designed as an iterative approach, the implementation allows for manual intervention, correction, or improvement of resources and parameters after each run without any limitations[19].

4.4 Chapter Summary

In this chapter, we have described a versatile approach to expand a lexical resource. The proposed approach is based on a model of distributional semantics which incorporates a quantifiable relation between terms in the sense of relative semantic similarity. In order to expand a lexicon, we leverage the properties of the embedding model and combine different representations (via vector addition) to create new candidates, given a lexicon and a search direction. We emphasize that even a small core (a couple of terms) is normally sufficient to quickly derive a lexical resource. We will report on the performance of the algorithm for different settings in Chapter 6 in order to illustrate its productivity and versatility.

But beforehand, in the next chapter we will turn to the challenge of applying such lexical resources to tackle the problem of skewness in the data distribution and the locality of information for classification tasks.

[19]See Chapter 6 for a list of examples. For instance the induction of terms about crimes in the financial sector (Section 6.2.2) contains a (scripted) change of the starting point and discusses also some erroneous inclusions during induction. However, these errors are counteracted with another strategy—also for reasons of reproducibility. But nothing prohibits the human interaction in any way.

5

Fine-Grained Classification Based on Concept Detectors

> "Necessity is the mother of invention."
> — Plato

```
In [136]: mod.wv.most_similar(positive=["Wort", "Konzept", "Algorithmus"], topn=10)
Out[136]:
[('Idee', 0.5828515291213989),
 ('Begriff', 0.5691336393356323),
 ('Strategie', 0.5526338815689087),
 ('Methode', 0.5514823198318481),
 ('System', 0.5466152429580688),
 ('Formel', 0.5418015718460083),
 ('Prinzip', 0.5382324457168579),
 ('Computerprogramm', 0.5202865600585938),
 ('entwickeln', 0.508611798286438),
 ('Plan', 0.5075653195381165)]
```

We have shown an approach to automatically derive lexical resources given a small seed for a concept in the last chapter. Now we turn to the question of how to apply such resources in order to tackle various concrete tasks.

As mentioned by (Grimmer and Stewart, 2013, p. 274), dictionary-based approaches have been used in the social sciences for a long time (cf. Stone et al. (1966)). They illustrate a common application scheme by explaining how such dictionaries are applied to classification problems. In typical cases, we create a score for a given dictionary for a given text. For each token in the text we perform a look-up in the dictionary. If the token is found in the dictionary, its weight is added to the score. In the case where we have dictionaries that contain words concerning a specific semantic concept that is be expressed via a scale (e.g. negative weights for words conveying negative tonality, positive weights for positive tonality), we may also combine these scores into one.

Grimmer and Stewart (2013, ibid.) point out that dictionary-based methods are simple and easy to apply for different classification problems for which off-the-shelf dictionaries (e.g., LIWC (Pennebaker et al., 2001)) may also be suitable. They also mention the possibility to derive such dictionaries from already coded (annotated) documents. However, they warn that cautiousness is required when applying dictionaries to other domains than those from which they have been derived. Hence, standard dictionaries often need to be adapted for the case at hand.[1]

For the approach that we describe in this chapter we will come from the opposite direction. We are also using lexical resources for the identification of a phenomenon, but we derive such customized resources for the problem at hand. This means, instead of applying any off-the-shelf dictionaries (let alone the weights of its entries), we create them with the algorithms described in Chapter 4. Furthermore, we naturally validate them by using annotated data as a proxy to estimate their aptness for the task at hand.

But there is an important point not made by Grimmer and Stewart (2013): a weakness of most dictionaries is that they are (in the described application scenario) brittle because of the rigid look-up of single tokens. In other words, the performance is directly dependent on the coverage of the vocabulary represented in the dictionary. As a consequence, in addition to the problem of ambiguous entries in the dictionary (creating possibly false positives), the mere lack of important terms which are present in the textual data but not in the dictionary causes false negatives. As mentioned, if there is enough annotated data, expansion techniques that aim at mitigating the vocabulary coverage problem

[1] Although we agree with Grimmer and Stewart that the validation of off-the-shelf dictionaries as well as problem-specific created ones is necessary, we do not agree with the "exceptional difficulties in validating dictionaries" they describe which mainly stem from the artificial granularity of the result, i.e., after the calculation of the scores for a document.

Trivially, investigating the scores produced by the application of a dictionary, the validation considers not the dictionary itself but its application. To assess the dictionary itself one would need to assess its components, i.e., the entries. However, if we are interested in the aptness of the dictionary for and within an application—as we will do in the following—using the proposed method from Grimmer and Stewart (2013) to compare the performance of dictionary-based applications to gold labels is sufficient in our view.

are directly applicable. On the other hand, if we expand the dictionary with these techniques, it will never cover common synonyms or related terms that were not seen during data annotation (or not yet known at all).

In the next section, we will show how we profit from a more sophisticated representation of the text and how we transform the lexical resources to efficiently apply them in the embedding space.

5.1 Intuition by Example

Let us suppose that we have a content analysis case at hand, where we would like to find texts in which the President of the United States personally talks to the media himself.

Now consider the two following sentences[2]:

- Der Präsident der USA redete mit den Medien. (*The President of the US talked to the media.*)
- Barack Obama sprach mit der Presse. (*Barack Obama spoke to the press.*)

Given our dictionary covered the terms *Präsident* und *Medien*, we could easily identify the first sentence as being an almost prototypical example for our case. But since we have only the generic form of the function (*Präsident*) in the dictionary, we would not be able to recognize that *Barack Obama* is an instance of the class of Presidents of the United States. Additionally, since we have *Medien* in our dictionary but not *Presse* (a specific subset of the media; linguistically a hyponym), we would fail to identify it with our lexicon. Also, the verb of communication is different (*reden (to talk)* vs *sprechen (to speak)*), which means also this closely related terms both have to be covered by the dictionary.

If we check in the embedding model the most similar terms to the candidates from our two sentences, we notice that we should be able to use this representation in order to generalize (consider Table 5.1, 5.2, 5.3).

It must be mentioned that the given example is a bit artificial because it is compiled for illustrative purposes, in order to refer to the potential of semantic abstraction by chaining three concepts: the president of the US, verbs of communication, and the media[3].

[2]This is example is minimally adapted from the one in Kusner et al. (2015).

[3]While the task of detecting this concrete situation with the given actors is not a typical case in the sense of classification for automated content analysis, it is a reasonable example. Especially with presidents that do not talk to the media anymore at press conferences.

5 most similar entries to "reden"		
Rank	Word	Similarity
1	**sprechen**	0.6656
2	diskutieren	0.6182
3	lachen	0.5847
4	verstehen	0.5589
5	nachdenken	0.5449

TABLE 5.1: 5 most similar terms to *reden* in the semantic space of the word2vec model, ordered by cosine similarity

5 most similar entries to "Presse"		
Rank	Word	Similarity
1	**Media**	0.7610
2	Zeitung	0.6356
3	Journalist	0.6067
4	Boulevardblatt	0.5720
5	Öffentlichkeit	0.5620

TABLE 5.2: 5 most similar terms to *Presse* in the semantic space of the word2vec model, ordered by cosine similarity

5 most similar entries to "Barack_Obama"		
Rank	Word	Similarity
1	Obama	0.8664
2	Obamas	0.8132
3	**Präsident_Obama**	0.7965
4	**US-Präsident**	0.7917
5	Hillary_Clinton	0.7866

TABLE 5.3: 5 most similar terms to *Barack_Obama* in the semantic space of the word2vec model, ordered by cosine similarity

Nevertheless, it hints at the potential applicability of the used representation. For more generic phenomena (in the sense of a category for a whole document (see Chapter 7) or the occurrence of a specific framing (see Chapter 8)) we make the same assumption: if we have a lexicon consisting of words that are approximately close enough to the concrete textual instantiation, we should be able to recognize and classify it.

5.2 Re-entering the Embedding Space

As we have seen in the given example, it would be of great advantage if we could use our resource in an application with an embedding based modeling. This means, considering the realm in which we would apply our lexical resources to analyze the content, we would prefer to actually perform the inference in the embedding space and not by comparing strings or chains of tokens with lists of strings. Rather, we would attempt to benefit from a semantic representation of terms in the form of an embedding so that it would allow us to profit from the generalization it offers.

This leads in turn to the question how we should embed our lexical resources. The most obvious way would be to embed each term in the given lexicon and use it further as a unit for comparison. But this would cause at least two problems: first, we would have to calculate the resulting label considering not only *one lexicon* but instead considering *one lexicon for each entry*! Second, the performance in terms of speed of such an implementation would also be problematic, given that we also want to include lexicons comprising up to several thousands of terms.

A natural step to reduce the number of embedded units which we have to compare to the text at hand is to cluster them. This is also a straightforward procedure for n-dimensional continuous vector representations which is what we have to cope with in the case at hand.

As it turns out, with simple k-means clustering, we do not only obtain the cluster membership of the embedded terms but also the centroid of the cluster in the semantic space. This is a point in the n-dimensional space for which the sum of the distances to the members of the cluster is minimal. It is therefore a good candidate to represent the cluster members in the embedded space.[4]

Since the centroids are neither necessarily terms from the lexicon nor from the vocabulary of the embedding model, the algorithm is given all the freedom it needs to minimize the within-cluster distances. This in turn allows us to find a point with high similarity to terms which belong to the cluster. In fact, with this procedure, we infer points in the semantic space which are closer than the nearest neighbor in the embedding model. And this point is not only the new nearest neighbor to one single term of the cluster members, but for most of them.

For a more detailed and insightful inspection, let us explore the semantic space for a simple investigation: consider the nearest neighbors for the terms[5] *Kaugummi (chewing gum)*, *Tiger (tiger)*, *Hand (hand)*, and *Schadenfreude (schadenfreude)* in Tables 5.4 to 5.7. As we see, the nearest neighbors have a similarity between 0.60 and 0.69 in the semantic space. If we simply calculate the mean of the given terms, we observe that the nearest neighbors to this point are indeed the terms given to calculate the mean (see Table 5.8). While the point resulting from taking the mean is actually the new nearest neighbor for *Hand* (0.63 vs. 0.61 to *Finger* in Table 5.6) and for *Kaugummi* (0.604 vs. 0.597 to *Zahnpasta* in Table 5.4), the resulting point is not the nearest neighbor for *Schadenfreude* (0.59 vs. 0.69) and *Tiger* (0.56 vs. 0.61).

[4]Given that this point was found iteratively reducing the inertia, or in other words, by minimizing the sum of distances between the cluster members and its centroid, we naturally observe high similarity values between the centroid and the cluster members.

[5]These terms are arbitrarily chosen for illustration purposes. However, the terms are *not* arbitrarily chosen in the sense that they consist of a group of terms that are from different domains.

5 most similar entries to "Kaugummi"		
Rank	Word	Similarity
1	Zahnpasta	0.5970
2	Schokoriegel	0.5834
3	Süssigkeit	0.5803
4	Bonbon	0.5783
5	Papiertaschentuch	0.5759

TABLE 5.4: 5 most similar terms to *Kaugummi* in the semantic space of the word2vec model, ordered by cosine similarity

5 most similar entries to "Tiger"		
Rank	Word	Similarity
1	Löwe	0.6098
2	SCL_Tiger	0.5657
3	Lions	0.5517
4	Elefant	0.5454
5	Raubkatze	0.5277

TABLE 5.5: 5 most similar terms to *Tiger* in the semantic space of the word2vec model, ordered by cosine similarity

5 most similar entries to "Hand"		
Rank	Word	Similarity
1	Finger	0.6125
2	Arm	0.5769
3	link_Hand	0.5715
4	Kopf	0.5687
5	Hosentasche	0.5530

TABLE 5.6: 5 most similar terms to *Hand* in the semantic space of the word2vec model, ordered by cosine similarity

5 most similar entries to "Schadenfreude"		
Rank	Word	Similarity
1	Häme	0.6917
2	Neid	0.6113
3	Spott	0.5945
4	Mitleid	0.5664
5	Verwunderung	0.5543

TABLE 5.7: 5 most similar terms to *Schadenfreude* in the semantic space of the word2vec model, ordered by cosine similarity

5 most similar entries to the mean of "Kaugummi", "Tiger", "Hand", and "Schadenfreude"		
Rank	Word	Similarity
1	Hand	0.6316
2	Kaugummi	0.6036
3	Schadenfreude	0.5861
4	Tiger	0.5615
5	Mund	0.5477

TABLE 5.8: 5 most similar terms to the mean of the vectors of *Kaugummi*, *Tiger*, *Hand*, and *Schadenfreude* in the semantic space of the word2vec model, ordered by cosine similarity

So we confirm that adding vectors—or taking the mean from them which is equivalent concerning cosine similarity—is an acceptable way to combine multiple words into one point. In other words, the resulting point in the semantic space is close to all of the given terms and is at the same time not closer to other terms.[6]

5 most similar entries to centroid 1		
Rank	Word	Similarity
1	Hand	0.7874
2	Kaugummi	0.7874
3	Finger	0.6146
4	Mund	0.6115
5	Taschentuch	0.6013

TABLE 5.9: 5 most similar terms to the centroid 1 of the cluster model for *Kaugummi*, *Tiger*, *Hand*, and *Schadenfreude* in the semantic space of the word2vec model, ordered by cosine similarity

5 most similar entries to centroid 2		
Rank	Word	Similarity
1	Tiger	0.7536
2	Schadenfreude	0.7536
3	Häme	0.5087
4	Löwe	0.4831
5	Mitleid	0.4745

TABLE 5.10: 5 most similar terms to the centroid 2 of the cluster model for *Kaugummi*, *Tiger*, *Hand*, and *Schadenfreude* in the semantic space of the word2vec model, ordered by cosine similarity

But if we instead allow for building two clusters out of the given terms and inspect the most similar points in semantic space to the resulting centroids, we are confronted with an interesting property (see Table 5.9 and Table 5.10). Because we have given the clustering model the freedom to build two clusters out of the four (non-similar; see Table 5.11[7]) terms, it has clustered them according to the distance in semantic space. More interestingly, the centroid for the given terms in each cluster is now much closer to those terms than the previously known nearest neighbors in the vocabulary of the embedding model. More precisely, the centroid of cluster 1 has a similarity of 0.79 to *Hand* as well as to *Kaugummi* while the centroid of cluster 2 has a similarity of 0.75 to *Tiger* as well as *Schadenfreude*.

How can a point be closer to *Tiger* than *Löwe (lion)* while being at the same time closer to *Schadenfreude* than *Häme (malice)*? There was by no means a change of the position in the semantic space from the given terms, i.e., *Tiger* and *Schadenfreude* are still as

[6]Due to the fact that the mean of all embedded terms given is a kind of a "compromise" of their respective representations, we also understand that this compromise cannot easily maximize all of its targets, so to speak. In consequence, the resulting similarity values are going to slightly decrease the more (unrelated) words we add up for the combination. In a similar fashion, the compromise gets also easily drawn towards a group of similar terms if existent in the given list of terms.

[7]Note that cosine similarity is symmetric, hence the matrix is also symmetric.

Similarity Matrix for "Kaugummi", "Tiger" "Hand", and "Schadenfreude"				
	Kaugummi	Tiger	Hand	Schadenfreude
Kaugummi	1.0	0.070	0.240	0.128
Tiger	0.070	1.0	0.132	0.136
Hand	0.240	0.132	1.0	0.133
Schadenfreude	0.128	0.136	0.133	1.0

TABLE 5.11: Similarity matrix for *Kaugummi*, *Tiger*, *Hand*, and *Schadenfreude* in the semantic space of the word2vec model, based on cosine similarity

dissimilar as in the beginning (cosine similarity of 0.14, see Table 5.11). But to quote Sahlgren (2006, p. 20, emphasis added):

> "[...] high-dimensional spaces behave in ways that might seem counterintuitive to beings such as us who live in a spatially low-dimensional environment. Even the most basic spatial relations—such as proximity—behave differently in high-dimensional spaces than they do in low-dimensional ones. We can exemplify this without having to plunge too deep into mathematical terminology with the simple observation that *whenever we add more dimensions to a space, there is more room for locations in that space to be far apart*: things that are close to each other in one dimension are also close to each other in two, and generally also in three dimensions, but can be prohibitively far apart in 3 942 dimensions."

On the other hand, this also means with the chosen dimensionality of 400 for the semantic space, the embedding model allows to find points which are counterintuitively close to otherwise distant points.[8] Additionally, the points that are found are only the means of the members of the cluster. Hence, in the given example the resulting centroids are the mean of *Kaugummi* and *Hand*, and of *Tiger* and *Schadenfreude*, respectively[9].

Now we will turn to examples of (longer) lists of terms which resemble more the nature of the concepts we try to identify with our lexicons. In those cases, the terms that we are clustering are rather closely related in respect to the semantic space. This is a natural outcome from the lexicon generation process, since it is based on the axis of semantic similarity (see Chapter 4).

Consider for example the resulting centroids with their nearest neighbors for a derived lexicon given the concept of space travel[10] given in Table 5.12.

[8] This is because there is enough room to be far apart from all the other terms in the space while still being a mixture of the given terms.

[9] Trivially, this is the case since the calculation of the centroid during the k-means clustering is exactly based on this procedure.

[10] The lexicon comprises of 225 terms. We will return to this category in the context of the empirical evaluation concerning document classification in Chapter 7.

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Cape_Canaveral	0.8492
2	Canaveral	0.8372
3	Weltraumbahnhof_Cape	0.8298
4	US-Raumfähre_Discovery	0.8264
5	Spaceshuttle_Discovery	0.7933
6	Kourou	0.7924
7	Weltraumzentrum	0.7794
8	russisch_Sojus-Rakete	0.7782
9	Weltraumbahnhof_Baikonur	0.7768
10	Raumfähre_Atlantis	0.7761

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Raumsonde	0.8687
2	Sonde	0.8644
3	europäisch_Raumsonde	0.8294
4	rot_Planet	0.8230
5	Nasa-Sonde	0.8165
6	Hubble-Weltraumteleskop	0.8088
7	Weltraumteleskop	0.8047
8	Nasa-Raumsonde	0.8039
9	Orbiter	0.8009
10	Sonde_Rosetta	0.7983

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Raumtransporter	0.8179
2	Automated_Transfer	0.7931
3	Trägerrakete	0.7922
4	unbemannt	0.7832
5	Ariane-5-Rakete	0.7765
6	russisch_Sojus-Rakete	0.7750
7	Dragon-Kapsel	0.7661
8	Forschungssatellit	0.7648
9	Telekommunikationssatellit	0.7607
10	Raumfähre	0.7595

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Himmelskörper	0.8852
2	Sonnensystem	0.8796
3	unsre_Sonnensystem	0.8778
4	Galaxie	0.8597
5	Komet	0.8550
6	Planet	0.8432
7	Asteroid	0.8288
8	Milchstrasse	0.8238
9	unsre_Milchstrasse	0.8132
10	Astronom	0.8112

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Weltraum	0.8539
2	Raumsonde	0.8373
3	Sonde	0.8228
4	Weltall	0.8210
5	Raumschiff	0.8163
6	All	0.8112
7	Umlaufbahn	0.8058
8	Satellit	0.8058
9	Raumfahrzeug	0.8001
10	Astronaut	0.7938

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	All_befördern	0.8354
2	All_schießen	0.8240
3	Forschungssatellit	0.8098
4	All_schicken	0.8054
5	Erdumlaufbahn_bringen	0.8047
6	Mondsonde	0.7944
7	Satellit	0.7775
8	Testsatellit	0.7759
9	Raumsonde	0.7660
10	Trägerrakete	0.7653

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	international_Raumstation	0.8881
2	Raumfähre	0.8741
3	Astronaut	0.8552
4	ISS_angedockt	0.8441
5	Raumfähre_Atlantis	0.8412
6	ISS_starten	0.8374
7	Endeavour	0.8338
8	ISS_fliegen	0.8281
9	international_Weltraumstation	0.8269
10	Cape_Canaveral	0.8266

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Jupiter	0.8711
2	Mond	0.8518
3	Mars	0.8330
4	Merkur	0.8329
5	Neptun	0.8218
6	Saturn	0.8179
7	Komet	0.8165
8	sonnennah_Planet	0.7880
9	Planet	0.7869
10	Planet_Merkur	0.7786

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Langstreckenrakete	0.8283
2	ballistisch_Rakete	0.7786
3	Interkontinentalrakete	0.7644
4	ballistisch	0.7524
5	Mittelstreckenrakete	0.7517
6	Missil	0.7505
7	Marschflugkörper	0.7422
8	Nuklearsprengkopf	0.7381
9	Rakete	0.7229
10	Raketentyp	0.7196

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Nasa	0.8790
2	ESA	0.8143
3	europäisch_Raumfahrtorganisation	0.8112
4	Esa	0.8027
5	amerikanisch_Raumfahrtbehörde	0.7943
6	US-Raumfahrtbehörde_Nasa	0.7931
7	Raumsonde	0.7878
8	amerikanisch_Weltraumbehörde	0.7845
9	Jaxa	0.7842
10	europäisch_Raumfahrtagentur	0.7777

TABLE 5.12: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for space travel in the semantic space of the word2vec model, ordered by cosine similarity

As a start, we observe again that the centroids from the different clusters show very high similarity values to their nearest neighbors in general: between 0.82 (centroid 3) and 0.89 (centroid 7). And also even for the 10^{th} most similar neighbor, the similarity is between 0.72 (centroid 9) and 0.83 (centroid 7). Thus, the resulting centroids are in "dense environments", which means that they are highly similar to many points (terms) of interest.

Putting the distances to the centroids under scrutiny from the perspective of the terms in the lexicon, further properties of the result of the cluster process are revealed. Out of the 225 terms in the lexicon, 64 are closer than 0.8 to at least one of the centroids in terms of cosine similarity. Even more than two third of the terms (152, i.e., 67.6%) are closer than 0.75. And if 0.7—which is still an arguably high similarity in comparison with other terms in the embedding model—would be the threshold, 92.4% (208 out of 225 terms) of the lexicon are not farther apart than this value. This means in turn that almost all of the given terms from the lexicon are well represented in the 10 centroids.[11] And as a consequence, we have thereby discovered a point in the semantic space which is for many terms their new nearest neighbor, using only 10 clusters. Hence, we would argue that the number of clusters is sufficient to represent the variety in the given lexicon.

	C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**
C1	1.0	0.767	0.849	0.524	0.796	0.819	0.887	0.553	0.571	0.799
C2	0.767	1.0	0.805	0.83	0.878	0.803	0.795	0.794	0.481	0.866
C3	0.849	0.805	1.0	0.588	0.876	0.88	0.879	0.587	0.628	0.868
C4	0.524	0.83	0.588	1.0	0.767	0.606	0.543	0.876	0.361	0.654
C5	0.796	0.878	0.876	0.767	1.0	0.876	0.861	0.741	0.587	0.84
C6	0.819	0.803	0.88	0.606	0.876	1.0	0.828	0.593	0.697	0.804
C7	0.887	0.795	0.879	0.543	0.861	0.828	1.0	0.568	0.515	0.839
C8	0.553	0.794	0.587	0.876	0.741	0.593	0.568	1.0	0.314	0.619
C9	0.571	0.481	0.628	0.361	0.587	0.697	0.515	0.314	1.0	0.505
C10	0.799	0.866	0.868	0.654	0.84	0.804	0.839	0.619	0.505	1.0

TABLE 5.13: Similarity Matrix for the 10 centroids of the cluster model for the lexicon for space travel in the semantic space of the word2vec model, based on cosine similarity

This is also interesting when we turn to the distances between the cluster centroid points. Table 5.13 gives an overview about the similarities of the centroids (C1-C10). Besides the table with the real-valued cosine similarities, we also provide a visualization for the centroids. In the PCA plot of the centroids in Figure 5.1 we already observe from the relative positioning that centroid 9 is far apart from the other ones. Additionally, we also notice that centroid 4 and centroid 8 are really close together (similarity of 0.88) and a bit farther apart from the other clusters.

Although these intuitive observations may be corroborated by the bare similarity values, the projection must be studied with care: the compression into two-dimensional space

[11] The average of the distances to the nearest centroid over all terms from the space travel lexicon is 0.770 with a standard deviation of 0.051

for the plot is misleading if we consider the difference in the distances.[12] Nevertheless, the real distances given in Table 5.13 show that they are substantial. Thus, although the lexicon contains many closely related terms, also sub-concepts are distinguished through the clustering. This holds true for centroids that are far apart (like centroid 9 which covers the cross section to military interaction with space travel), as well as for more similar clusters (centroid 4 and 8 which are both related to planets).

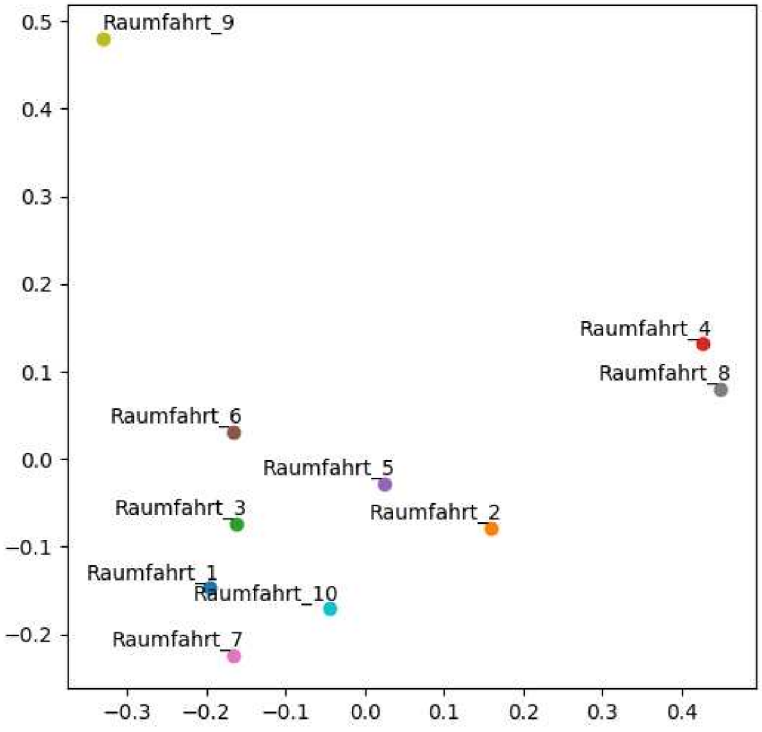


FIGURE 5.1: PCA projection of the centroids for the space travel lexicon

These observations are in correspondence to a closer investigation of the clusters, referring again to the given nearest neighbors in Table 5.12. If we have to describe the captured sub-concept for each cluster, we may identify places for rocket and space shuttle launches in cluster 1. Cluster 2 covers devices that have already been sent into space (for instance to Mars). Even more specific rocket types are covered in cluster 3. Space *per se* as well as space travel vehicles are grouped in cluster 5. Cluster 6 is orbiting around the procedure to place satellites and other devices in space. By contrast, in cluster 7, we find multiple references to the International Space Station(ISS) and the space shuttles supplying it.

[12]The cumulative explained variation for the first two principal components is only about 0.66.

Now with a bit of a subtlety, the mutually close clusters 4 and 8 contain similar reference points. But while centroid 8 includes the single instances of the planets of the solar system, cluster 4 is more general in the sense that it features more generic names for celestial bodies (including *Himmmelskörper (celestial body)* itself) and even references to the Milky Way, as well as galaxies in general. Note that *Planet* is in both lists of the 10 most similar terms. This is not an error or a noisy inclusion. *Planet* is just what is present in both clusters prominently. Once as a (generic) instance of a celestial body, and once rather as a hypernym for all instances. This example points nicely to the way in which ambiguity of terms is included into the centroids during the re-embedding and clustering process.

While cluster 10 clearly refers to the organizations and agencies for space travel, cluster 9 is the outlier considering the focus on military weaponry. This is also due to the German word *Rakete* that may refer to *missile* or *rocket* which caused the weakly supervised lexicon generator to include questionable terms which in turn were taken into account by the clustering mechanism. This cluster is shown intentionally to illustrate the following point: while the clustering process is reasonably robust, it is still attentive to special sub-concepts in the lexicon if they are distant to the others. This happens especially if the variance between the other clusters is rather low and due to the assumption of k-means clustering (similar size of clusters and spherical data distribution). In this way, distant clusters for a few words differ substantially from the rest are induced.

Summing up, the simple method of re-embedding and k-means clustering offers two things: first, the terms in the lexicon are grouped based on their similarity in the embedded space, hence according to their meaning. Second, due to the properties of the clustering algorithm, the centroids built during the iterative reduction of inertia are a recombination of the cluster members. This is especially useful for terms with similar meaning, since that will result in cluster centroids which have a high similarity to all the members of the cluster. In addition to the meaningful sub-clustering of the given concept (or the lexicon respectively), we also detect further terms that reflect the specific meaning of the lexicon sub-cluster (see also Chapter 6). This information is accessible for inspection by checking the nearest neighbors of the resulting centroids.

We would like to emphasize that we do not claim that the quantization using k-means clustering is an optimal approach. Rather, we have shown empirically that clustering yields robust results for a meaningful representation of the lexicon (see Section 6.4.1 and 6.4.2.1) which we further integrate to build a classifier (see Chapter 7 and 8 for the extrinsic evaluation in the respective tasks.). We will now elaborate further how to use these embedded lexical resources in order to create a versatile yet competitive classification system. From now on, we will also refer to the centroids used in the

classifier as detectors, since they will give an amount of signal (i.e., similarity) given any embedded input.

5.3 A Fine-grained Classifier

As we have delineated the way to transform the lexical resources into detectors (centroids of a clustering process to quantize the re-embedded lexicon), we will now describe a simple classifier based on those detectors. The proposed implementation is geared towards the goal to be as simple as possible whilst still performing well. In the next section, we first briefly explain again how we model the input data for classifier.

5.3.1 Unit of Analysis

The projection of the lexical resource into n centroids or detectors is only one part of the application. We also need to find ways to represent the textual input so that the comparison yields reliable results while the choice for the *unit of analysis* offers enough leeway for adapted modeling that fits the content analysis at hand.

In most cases, the actual unit of analysis is given through the design of the analysis and in many content analyses the unit of analysis will be a document or an article. We have seen in the last sub-chapter that taking the mean of a set of (embedded) given terms results in a representation closely to all the given terms (cf. Table 5.8). But as pointed out earlier, this kind of combination of the terms into a single point has its limits.

First, using too many different terms in one combination results in generally decreased proximity to the terms (cf. Table 5.14 which contains three additional non-similar terms). Second, some of the terms are not represented equally; *WEF (World Economic Forum)* for example has only a moderate similarity of 0.43 to the new mean of all terms, resulting in 35th rank. Additionally, it is not obvious how to determine which term will not be represented well by the pure mean of all vectors a priori.[13]

Similarly, if the list contains terms which are more similar to each other than the others, they tend to dominate the combination. Let us for example add the terms *Van_Gogh* and *Pinsel (brush)* to the original quadruple.

In Table 5.15 we observe how the similarities of the single terms to the resulting point now differ with considerable variance. While the resulting point is very similar to *Pinsel*

[13]For the given example, where intentionally unrelated terms were mixed together, it is actually hard to argue why one term is represented worse than others. This issue is also persists when we move to the clustering process and project the given set of terms into more than one point.

Similarity to a combination of "Kaugummi", "Tiger", "Hand", "Schadenfreude", "WEF", "Paradoxon", and "Microsoft_Word"		
Rank	Word	Similarity
1	Kaugummi	0.5366
2	Tiger	0.5020
3	Hand	0.5001
4	Microsoft_Word	0.4995
6	Schadenfreude	0.4916
8	**Paradoxon**	**0.4814**
35	**WEF**	**0.4322**

TABLE 5.14: Similarity for seven given terms to the mean of their vectors in the semantic space of the word2vec model, ordered by cosine similarity

(0.70), the similarity values to *Schadenfreude* and *Tiger* drop below 0.46. If we additionally consider the rank in the list of nearest neighbors, the situation is even clearer with ranks 199 and 254 for *Schadenfreude*, and *Tiger*, respectively.[14]

Similarity to a combination of "Kaugummi", "Tiger", "Hand", Schadenfreude", "Van_Gogh", and "Pinsel"		
Rank	Word	Similarity
1	Pinsel	0.7000
2	Hand	0.6012
3	Kaugummi	0.5965
6	Van_Gogh	0.5426
199	**Schadenfreude**	**0.4580**
254	**Tiger**	**0.4524**

TABLE 5.15: Similarity for six given terms to the mean of their vectors in the semantic space of the word2vec model, ordered by cosine similarity

If we intend to represent whole articles or documents as units of analysis, we may assume that a simplistic modeling based on the full unit of analysis, (i.e., projecting the whole text into a single point in the embedding space) would lead to considerable inappropriateness.[15] On the other hand, we have observed that a limited number of terms is nicely represented in one point by taking the mean from them. And if the terms are from a specific domain, the combination will probably yield really high similarity values.

The suggested solution here is to model the input into smaller units, so that the modeled state is reliably covering the input. This means that instead of embedding a whole input text into one point, we propose to do so on the level of sentences. Of course, as a

[14] Additionally, it is not intuitively clear why *Van_Gogh* is still considerably less similar to the resulting point (very close to *Pinsel*) than *Kaugummi*. When put under scrutiny, this is mainly because the similarities to *Pinsel* from *Van_Gogh* and *Kaugummi* are both 0.39. This is not to say that this is perspicuous—this justification is based purely on the simple math of the embedding model, and, thereby, pointing again to the unpredictable subtleties of embeddings.

[15] One might be reminded of the decreasing quality of bag-of-words modeling approaches given longer documents containing multiple yet different topics.

consequence, such a modeling requires us to formulate a way to compose the sentence-level analysis into a document-level analysis, and, of course, also one to compose a word-level analysis into a sentence-level analysis.

However, this leads to a second advantage of the modeling of smaller scopes: the interesting piece of information for content analysis in social sciences is often not distributed and prevalent over the whole document. On the contrary, the important parts of the texts are often strongly locally bounded, so that the focus to recognize them must be narrow (see also Chapter 8). Otherwise the information could get lost due to a modeling that mixes the contents of the whole text. But if the analysis is carried out on the sentence-level, we strive for detecting those small yet most important parts of the text.[16]

At the same time, we will have a disassembled fine-grained analysis at hand for the document, since we will have to compose a document-level analysis out of sentence-level analyses anyway. And since we apply the beforehand described detectors that further model the lexical resources into sub-concepts, we even get a per-sentence analysis on the sub-concept level.

Naturally, this comes at the cost that we predict in principle for each sentence its property to refer to a given phenomenon that is maybe also distinguishable on the document level. But given the necessity to narrow the scope because of the properties of the modeling approach (embedding the input text) and the expected advantages of the fine-grained analysis—which also copes with locally bounded information—we consider these additional costs as an acceptable trade-off.

5.3.2 Algorithm

As we have discussed in the preceding section, we will model the input on the level of the sentence. And as we illustrated beforehand, we will use the quantized version of the lexicons (i.e., reduced through clustering to n centroids, or detectors). While the unit of analysis for the classifier may well be a document, we will actually work on the level of sentences to measure the similarity between the input and the lexical resources.

Again, like in Chapter 4 we will provide a verbalized version of the process as well as an algorithm in pseudo-code subsequently. While the first will contain more examples and details, the latter will be denser and more formal.

[16] Of course, if we have to identify a phenomenon which is *only* perceivable on the supra-sentence-level, one would have to adapt the modeling to the case at hand. If enough annotated data is available, it may be worthwhile to consider methods like `doc2vec` (Le and Mikolov, 2014) which embeds whole paragraphs or documents. However, it remains unclear how the analogous lexical resource would look like if we compare it to our approach using word embeddings.

The following steps are carried out during processing:

1. The input text is split into sentences. For each sentence, we create an embedded version, i.e., we calculate a point in the embedded space that represents the content of the sentence. For this calculation, we additionally apply a filter based on part-of-speech tags (for simplicity, the default filter embeds only nouns and adjectives). The embedded version of the sentence is thus the mean of the filtered lemmata of the sentence.

2. Each of the lexicons is clustered beforehand and the m detectors for each concept are saved. To prepare the classifier, we load them into a matrix of detectors.

3. Each embedded sentence is compared to the matrix of detectors in terms of similarity. This results in a matrix of similarities per sentence.

4. The matrices containing the similarity values are evaluated. Two parameters exert an influence on the assessment: the threshold t and the n-best value. The threshold t is applied to the similarity values. It has empirically turned out that a default value of $t = 0.3$ filters out detector signals that result from the fact that also very noisy signals add up to a "base similarity" in the embedded space. For all the detector signals higher than the threshold, we apply a second filter: we take only the n-best of them into account (default for $n = 10$). If fewer than n detectors have passed the first threshold, all of them are taken into account.

5. The filtered signal values (i.e., the similarity values) from all detectors are aggregated per concept. This means that each detector contributes now to the concept it stems from. We then compare the summed signals over the different concepts (summed values or as percentage over all summed values). For single label prediction, we now choose the highest scoring concept (category). The distribution of the signals across all categories is also given.

Listing Algorithm 3 gives a more compressed overview using pseudo code which is equivalent to the respective verbal step-by-step description. We describe the prediction for one input text.

Algorithm 3
Classification Based on Embedded Lexical Resources

Input: Embedding E, Detectors D, Parser P, Text T; optional: Filter F, Threshold t, Cut-off: n-best
Output: Prediction of Category C; Distribution over Categories C_{dist}

$\varphi(Sim, t) = \{x \mid x \in Sim \wedge x > t\}$ ▷ similarities above threshold
$S \leftarrow$ sentence_split(T, P) ▷ splitting text into sentences
for *sentence* $\in S$ **do**
  $e_{sentence} \leftarrow$ embed_sentence$(sentence, E, F)$
  $SimMat_{D\,sentence} \leftarrow e_{sentence} \oplus D$ ▷ $\oplus$: cosine similarity
  $sims\_above\_threshold \leftarrow \varphi(SimMat_{D\,sentence}, t)$
  if $|sims\_above\_threshold| > n$ **then**
    $top\_n\_scores \leftarrow \underset{top\ k=n}{\text{argmax}}\ \varphi(SimMat_{D\,sentence}, t)$ ▷ filter for n-best
    $concept\_score\_list \overset{+}{\leftarrow} top\_n\_scores$
  else
    $concept\_score\_list \overset{+}{\leftarrow} sims\_above\_threshold$
$C \leftarrow$ argmax aggregate$(concept_score_list)$ ▷ summing over detectors per concept
$C_{dist} \leftarrow$ aggregate_percentage$(concept_score_list)$ ▷ percentage over sums
return C, C_{dist}

5.3.3 Parameter Discussion

In this section, we briefly discuss the two parameters used in the classification approach. Additionally, we point to the (weak) interdependence of the parameters to the number of concepts or categories and how this should be taken into account while setting parameter values for the case at hand.

5.3.3.1 Threshold for Similarity

The threshold for the similarity value of the detectors to be taken into account has a default value of $t = 0.3$. This default value is set rather low so that the detectors also capture weak signals. If it is increased, there will be no signals above t for many sentences (let alone more than n detector signals). This in turn means that we consider those sentences as not being informative enough given the concepts realized as detectors.[17] However, it may turn out to be a useful filter, if the concepts which we want to distinguish are closely related and therefore we want to filter for really clear-cut signals. But this is also related to the number of concepts (see below).

[17]Note that this is often intentionally the case. Since we know that the information we would need to pay attention to is often locally bounded, we do not suppose that each sentence contains evidence that points towards the phenomenon under investigation. Thus, we would rather ignore the assumedly non-contributing sentence than enforce a (doubtful) decision without firm grounding.

During extensive experimenting it turned out that this parameter is also especially useful for fine-tuning the classifier in cases where we need to focus on locally bounded information (see Chapter 8).

5.3.3.2 Selection of n-best Candidates

First, it must be noted that the restriction to n detector signals is applied as a second stage filter. It only crops the result set if the threshold t leads to more than n candidates. The main intention behind this second filter is to level out the inequality of contribution weight which may occur if a sentence is by whatever reason highly similar to a large number of detectors. To limit the influence with which such a sentence contributes to the overall prediction, we restrict the potential informative signals to n. Also, by limiting the potential signals taken into account to n, we introduce a generally normalizing factor prior to the aggregation over all sentences. The default value for n is 10.

5.3.3.3 Number of Detectors and Number of Concepts

Although not directly a parameter of the algorithm, the number of detectors plays an important role. If we suppose that we want to classify documents into one of three categories which we capture with three concepts, using ten detectors for each concept, setting n as high as 30 leads to a "use-all" mode and the second stage filter becomes useless.[18]

Additionally, if we have numerous categories that we measure with concept detectors, the threshold t should not be too low, so that we do not capture too many possibly false positive signals. In other words, since the chance to create erroneously signals above threshold increases naturally with the number of detectors, we should set the parameters a bit more restrictive when working with dozens of categories.

For a closer investigation of the interplay between the parameters, see also Chapter 7

5.4 General Remarks

In this section, we briefly summarize the proposed approach for the classifier and reflect on the steps taken.

[18]This is not to say that this setting is not useful by default. The point is that one should be aware that the number of detectors is directly influencing the way the filters work.

The main question of this chapter was how to integrate the lexical resources into an application framework based on embeddings. We have shown a method to embed the lexical resources, and, more importantly, quantize the lexicons in the sense that we cluster them. By inspection of the vicinity of the resulting centroids from the clustering process we find that they represent meaningful sub-concepts.

The control for the validity of the generated detectors (i.e. the points found as centroids in the semantic space) is crucial:

- This inspection reveals in general if the concept instantiated by the given lexicon is adequately represented using the distributional semantics of the embedding.
- If several centroids represent almost the same parts of the lexicon (quantified by the overlap of the n most similar neighbors of the centroids), this is a strong indicator that the number of clusters could be helpfully decreased. In other words: the given freedom to represent subtle differences within the concept is not usefully exploited.
- On the other hand, a centroid that represents heavily mixed semantic sub-concepts is an indicator that the number of centroids should be increased in order to cope with the variety given the content of the lexicon.
- Although the optimal number of clusters for the quantization process depends on the case at hand, we found during experimentation that the influence is negligible on the downstream classification process.
- However, if one has to deal with several lexical resources that differ in orders of magnitude considering size (e.g., dozens of words vs. thousands of entries in two supposed lexicons), we would recommend to use an adaptive number of clusters according to the lexicon size.[19]
- Additionally, the closer investigation of the centroids also allows to find problematic cases which stem for example from the ambiguity of lexicon entries. It remains in the hands of the application designer to decide if such centroids should be just ignored or need to be adapted (see Chapter 9) in order to reach the best performance. Other ways to surpass this problem are to experiment with the number of clusters and/or combine several clustering runs to find a more robust solution, or to simply edit the lexicon itself.

[19]To counterweigh the influence of the number of detectors, one may multiply the respective concepts inversely with the number of detectors. This is also an usable scheme for hierarchical classification settings. See Chapter 8 for an example where several concepts contribute to a super-concept.

To apply these embedded resources, also the input needs to be brought to the same realm, i.e., the embedded space. To retrieve an adequate representation using a simple composition scheme, we use the sentence as unit of analysis for the implementation of a classifier. This means we analyze the strength of the signals of the detectors for each sentence separately. While we must formulate a projection of these sentence-level analyses to the real unit of analysis (e.g. the paragraph, or the whole document), this gives us also the freedom to apply any kind of weighting scheme—be it learned from accessible data or fine-tuned according to other requirements like settings oriented towards recall or precision.

As mentioned above, this trade-off is attractive or at least acceptable, given the fine-grained measurement we receive in turn: a score of similarity of the sub-concepts of the lexicon(s) per sentence.

These detector signals are *per se* independent of each other, which is one of the noticeable differences to the classic supervised classification settings. Here, we do not try to find a separating hyper-plane in the n-dimensional space to discern the different classes. Instead, we first create an abstract view of the text via the signals of the detectors. Of course, such a view is not yet a classifier. Rather, it is comparable to a step of (high-level) featurization. Hence, we have proposed this simple schema for deriving a classification decision for single-label classification or to create the distribution across multiple given labels.

But the main advantage of such a setting is the absence of the requirement to explicitly define a category comprising of a set of negative examples in contrast to the phenomena of interest. That means, the "*OTHER*" class, or the residual category, does not have to be represented in the training data at all (although good examples that provide more clarity regarding the thresholds for the signals may be valuable).

This is especially useful if the residual category is much more prevalent than the phenomena to measure (see also Chapter 8), because especially in such settings, it is notoriously difficult to represent this category in its variety adequately[20] without the risk to over-generalize this class. Specifically in the case of over-generalization of the residual class, the problem to detect instances of small classes (which are according to the given distribution represented poorly in the training data) is aggravated. Since the proposed approach is based on the assumption that it is known beforehand what should be found (detected) conceptually, the definition of instances showing the absence of it is not a (formal) *conditio sine qua non* for the classifier itself. In other words, we are able to

[20] Of course, this category normally does not represent the *presence* of a concept anyway, but rather the *absence* of any of the concepts of interest.

build the classifier having only positive examples or even just a conception that suffices to profit from the lexicon generation algorithm delineated in Chapter 4.

Similarly, since the classification is based on externally generalized resources (through the distributional semantics backing the lexicon induction process), this also holds true for data skew problems *amongst* the given categories. In other words, given a skewed distribution of the phenomena under investigation, we counterweigh the influence of the overly represented categories by normalizing the sensitivity for each phenomenon by the number of detectors. While normally the improvement of the detection over *all* classes (e.g., in the sense of macro-recall; see Chapter 7) is in focus, the exact same axis of influence is also usable to tweak the classifier performance in any other direction.

Lastly, we would like to emphasize the simplistic parametrization, including only a threshold and a number for n-best. While it is clear that this choice of modeling in the solution we propose—which is intentionally kept simple—leaves much room for optimization on different levels, it yet provides a level of control to implement solutions adhering to specific desiderata (for the robustness of the valid parameter range, see also Chapter 7). Furthermore, this approach follows several guidelines geared towards transparency, inspectability, and simplicity. Introducing more parameters also tends to induce a higher level of opaqueness of the mechanics of the applied solution.

5.5 Chapter Summary

In this chapter, we have shown how we embed the lexical resources in order to apply them in a modeling that is based on the word embeddings and thus allows for generalization. In other words, we have described how and why we apply clustering methods to quantize the lexical resources in the semantic space and how this approach is further feasible to make use of the general semantics incorporated in the embedding model while still preserving attentiveness to the lexical resources.

We also introduced a method to implement a classifier based on this modeling and those resources. The proposed classification approach moves the decision point down to the sentence-level, mainly because of modeling restrictions. But the gains we get in turn are twofold: first, we get a per-sentence analysis on the sub-concept level (realized by the detectors). Second, this fine-grained analysis allows for versatile modeling, coping with challenging scenarios like detection of locally strongly bounded phenomena.

While the classification of the original unit of analysis (e.g., the document) requires a formulation of a projection from the sentence-level to the document-level, this is an

acceptable trade-off given the representational power (in the sense of semantic generalization) of the embedded space and the applicability of externally extensible and malleable lexical resources (see Chapter 4).

The empirical evaluation of the approach will be reported in Chapter 7 and 8.

6

Experiments I: Lexicon Induction

> "Get your facts first, then you can distort them as you please."
> — Mark Twain

```
In [31]: analogy(a="grau_Theorie", b= "Praxis",
x="Hypothese", y=None, model_given=model, verbose=True)
'grau_Theorie' is to 'Praxis' as 'Hypothese' is to 'Befund'
Out[31]:
[('Befund', 0.5098408460617065),
 ('Theorie', 0.5060617327690125),
 ('These', 0.4840780198574066),
 ('Methode', 0.47375476360321045),
 ('Beobachtung', 0.45115962624549866),
 ('Vermutung', 0.44649460911750793),
 ('Aussage', 0.4378140866756439),
 ('Einschätzung', 0.42812588810920715),
 ('Erkenntnis', 0.4267434775829315),
 ('Diagnose', 0.4163473844528198)]
```

In this chapter, we aim to evaluate the lexicon induction module outlined in Chapter 4. More precisely, we illustrate in detail how the iterative induction is carried out and therefore present an introductory example. In a second experiment we illustrate how the adaptation of a resource may be carried out. We therefore study two cases.

Firstly, we demonstrate how we identify specific verbs of communication conveying negative sentiment when starting only with a small seed set of general verbs of communication. This is an instance that illustrates the combinatory opportunities on the concept level. Secondly, we turn to a case where we investigate how to identify domain-specific criminality terms (from the financial sector). Based on the injection of general domain terms we adapt and expand a lexicon of crime terms. This case demonstrates how a lexicon that is based on "generic" terminology is adapted to a specific target domain.

Furthermore, we report on results from a third experiment, where we induce a sentiment lexicon from a small seed. This setting is chosen to estimate the quality of the outcome and also to demonstrate that the process may also be conducted with less supervision. On the one hand, we evaluate the produced sentiment lexicon manually and compare the expansion with a large pre-existing lexicon, as well as reporting on the applied techniques to further automate the creation of such resources. On the other hand, we do not evaluate the resource extrinsically through the performance of a downstream application.

Lastly, we also report on the lexicons we derived for specific tasks, namely document classification for small and imbalanced data sets (see Chapter 7) and for the detection of frames of democratic legitimacy (see Chapter 8). This last task also requires coping with heavily skewed data distributions. Since we focus in this chapter on the description and detailed consideration of the created resources, we leave the evaluation of the usefulness and efficacy of these resources to their respective chapters.

6.1 Intuition by Example: Dogs in the Embedding

For ease of understanding, we start with a simple example by inducing a lexicon of names of dog breeds. This is not related to the following experiments given in this section. We illustrate the lexicon induction algorithm's mode of operation on a simplified problem[1]. For this case, we show in detail which terms are taken into account as candidates during the search phase, and we report the result of the algorithmic assessment of the candidates. Additionally, the status of the lexicon and the shadow lexicon is reported.

[1]While the task of collecting names of dog breeds may be better accomplished by looking them up in an apt resource such as `https://en.wikipedia.org/wiki/List_of_dog_breeds`, we chose this setting because it is easy to understand. Nevertheless, it is interesting and certainly not self-evident that embeddings from media texts contain so much information about dogs.

The initial lexicon contains seven randomly chosen dog breeds (*Berner_Sennenhund, Chihuahua, Chow_Chow, Collie, Labrador, Schäferhund, Setter*) with no other prerequisites than the presence of the terms in the vocabulary of the embedding[2]. The following parameters have been set[3]:

- Number of runs: 3
- Number of iterations: 3
- Number of known terms to sample: 1
- Number of top known terms to sample from: 4
- Number of new terms to sample: 1
- Number of top new terms to sample from : 2
- Result Size: 50
- Shadow Lexicon Weight: 0.5
- Assessment Threshold: 0.1

Except for the number of runs and the number of iterations, all other parameters are set to their default values. To clarify the influence of the chosen parameters, we quickly summarize here: Initially, the first starting point is set. Since we do not give a defined starting point by providing one or more initial "steering terms"[4], we will get two terms randomly drawn from the lexicon (in the reported case, these terms are *Schäferhund* and *Labrador*). We start the first of three runs from each of which consists of three iterations. The first iteration starts with the collection of the 50 most similar terms to the starting point, which are—in the descending order of cosine similarity—handled as follows:

a) If a term is in the exclusion lexicon, it is not taken into account and discarded for any further processing right away.
b) If a term is in the lexicon itself, it is not a new candidate. But it is collected in the given order to be available for the re-sampling process.
c) If a term is not in the lexicon nor the exclusion lexicon, it is considered as a new candidate. Therefore, it is collected in the given order in the list for new candidates of the respective iteration.

[2]Additionally, we make use of a short exclusion lexicon, consisting of *Büsi, Frauchen, Gassi_gehen, Halter, Halterin, Herrchen, Katze*. These are terms which have a high similarity to our target (dog breeds) and hence appeared repeatedly in the result. Nevertheless, we would like the search no to be mislead in the direction of other animals or the owner of the dogs.

[3]The number of iterations per run has been decreased to 3 for illustrative purposes, i.e., to keep this already long step-by-step example a bit shorter. The reduction of the iterations has mainly an effect on the robustness of the search. However, the results in this introductory case study do not differ largely from those with a default parametrization, since this is a rather simple example.

[4]These terms define where the search for the lexicon induction starts. We refer to those as *starting point* in 4.2.5.1.

For the next iteration, the starting point will now be set according to the parameters. In the given case, we sample one from the top four terms in the list of known terms, i.e., terms that we already have in the lexicon. This element binds the next search step to the "concept of the lexicon", i.e., the idea which is captured by the collection of terms forming the lexicon. Additionally, we also sample one term from the top two terms of the list of the new candidates (terms absent from the lexicon). This element ensures that the search remains on the chosen path, initialized by the starting point. With these re-sampled terms (two in our case), we will create the next starting point (i.e., the terms will determine the new starting point by vector addition).

After the next two analogous iterations in the first run (initialized with the same starting point), we evaluate the collected new candidates. The terms are ordered by their frequency (i.e., in how many iterations they appeared in the new candidates). The rationale behind this ordering is that terms which show up in many iterations have a higher probability to be apt candidates. The number of the terms that are algorithmically evaluated is limited by the result size parameter which in our case is set to 50.

For the assessment of the terms, we use the default weighting schema for the shadow lexicon (0.5) and the default assessment threshold (0.1). The lexicon and the shadow lexicon get updated accordingly, i.e., each time the assessment threshold is met, the term is added either to the lexicon or the shadow lexicon. After the assessment of all candidates (limited by the result size of 50), we start the next run.

In the following we report comprehensively the steps described above with all the intermediate results. We underline the terms which have been chosen by the algorithm for the re-sampling of the new starting point for the next iteration.

Status before start:

- **Lexicon**: *Berner_Sennenhund, Chihuahua, Chow_Chow, Collie, Labrador, Schäferhund, Setter*
- **Shadow Lexicon**: *[]*
- **Exclusion Lexicon**: *Büsi, Frauchen, Gassi_gehen, Halter, Halterin, Herrchen, Katze*

Search for run 1:

- **Run 1, Iteration 1**:

- **Terms for the starting point**: *Schäferhund, Labrador*
- From top 50 nearest neighbors to the starting point:
  - **New candidates**: *Hund, Hündin, Golden_Retriever, Rottweiler, Dackel, Terrier, Dobermann, Vierbeiner, Pitbull, Jack_Russell, Bulldogge, Kampfhund, Riesenschnauzer, angeleint, Bullterrier, Pitbulls, Familienhund, Border_Collie, American_Staffordshire, Labradore, belgisch_Schäferhund, Mischling, Mischlingshund, Staffordshire_Terrier, deutsch_Schäferhund, Pudel, Labrador_Retriever, Hündchen, totbeißen, Hundehalter, Retriever, Dogge, Leonberger, Spaniel, Tierheim, Pitbull-Terrier, Welpe, Rehpinscher, Lein_führen, Rüde, Cockerspaniel, Terriers*
  - **Matches with lexicon**: *Collie, Berner_Sennenhund*
  - **In exclusion list**: *Herrchen, Katze, Frauchen, Büsi, Halterin, Halter*

- **Run 1, Iteration 2**:
  - **Terms for the starting point**: *Berner_Sennenhund, Hündin*
  - From top 50 nearest neighbors to the starting point:
    - **New candidates**: *Hund, Golden_Retriever, Welpe, Jack_Russell, Vierbeiner, Rottweiler, Border_Collie, Dackel, Leonberger, Appenzeller-Mischling, Familienhund, Rüde, Hündchen, Terrier, Leika, Mischlingshund, Schäferhündin, Pony, Riesenschnauzer, Mischlingshündin, Haustier, belgisch_Schäferhündin, Kätzchen, Pudel, Tierheim, Gismo, Schäfermischling, Meerschweinchen, Labradorhündin, sehr_anhänglich, deutsch_Schäferhund, anhänglich, verschmust, Pitbull, Kälbchen, belgisch_Schäferhund, Cockerspaniel, Zwergpinscher, Bernhardiner, Labradore, Bergamasker, Meersäuli, Manoi*
    - **Matches with lexicon**: *Schäferhund, Labrador, Collie*
    - **In exclusion list**: *Herrchen, Katze, Frauchen, Büsi, Halterin, Halter*

- **Run 1, Iteration 3**:
  - **Terms for the starting point**: *Schäferhund, Golden_Retriever*
  - From top 50 nearest neighbors to the starting point:
    - **New candidates**: *Hund, Rottweiler, Hündin, Terrier, Dackel, Pitbull, Vierbeiner, Kampfhund, Bullterrier, Pitbulls, Dobermann, Jack_Russell,*

American_Staffordshire, angeleint, Riesenschnauzer, Staffordshire_Terrier, Border_Collie, Labradore, Bulldogge, Leonberger, Familienhund, Rehpinscher, totbeißen, belgisch_Schäferhund, Mischlingshund, Pitbull-Terrier, Pudel, deutsch_Schäferhund, Labrador_Retriever, Hündchen, Cockerspaniel, Rüde, Dogo_Argentino, Lein_führen, bei_Gassigehen, Tod_beißen, Hundebesitzer, Mischling, Dogge, Hundehalter, American_Staffordshire-Terrier

- **Matches with lexicon**: *Labrador, Collie, Berner_Sennenhund*
- **In exclusion list**: *Herrchen, Katze, Frauchen, Büsi, Halterin, Halter*

Assessment of candidates from run 1

- **Candidates (with counts)**: *Hund(3), Rottweiler(3), Dackel(3), Terrier(3), Vierbeiner(3), Pitbull(3), Jack_Russell(3), Riesenschnauzer(3), Familienhund(3), Border_Collie(3), Labradore(3), belgisch_Schäferhund(3), Mischlingshund(3), deutsch_Schäferhund(3), Pudel(3), Hündchen(3), Leonberger(3), Rüde(3), Cockerspaniel(3), Hündin(2), Golden_Retriever(2), Dobermann(2), Bulldogge(2), Kampfhund(2), angeleint(2), Bullterrier(2), Pitbulls(2), American_Staffordshire(2), Mischling(2), Staffordshire_Terrier(2), Labrador_Retriever(2), totbeißen(2), Hundehalter(2), Dogge(2), Tierheim(2), Pitbull-Terrier(2), Welpe(2), Rehpinscher(2), Lein_führen(2), Retriever(1), Spaniel(1), Terriers(1), Appenzeller-Mischling(1), Leika(1), Schäferhündin(1), Pony(1), Mischlingshündin(1), Haustier(1), belgisch_Schäferhündin(1), Kätzchen(1)*
- **Added to lexicon**: *belgisch_Schäferhündin*
- **Added to shadow lexicon**: *Mischlingshund, Mischling, angeleint, Dobermann, belgisch_Schäferhund, Kätzchen, Border_Collie, Welpe, deutsch_Schäferhund, Spaniel, Riesenschnauzer, Dackel, Pudel, Labrador_Retriever, Terriers, Labradore, Rüde, Jack_Russell, Appenzeller-Mischling, totbeißen, Mischlingshündin, Bulldogge, Retriever, Dogge, Familienhund, Golden_Retriever, Lein_führen, Leonberger, Hündin, Schäferhündin, Leika, Hündchen, Rehpinscher, Cockerspaniel*

Status before run 2:

- **Updated Lexicon**: *Berner_Sennenhund, Chihuahua, Chow_Chow, Collie, Labrador, Schäferhund, Setter, belgisch_Schäferhündin*

- **Updated Shadow Lexicon**: *Appenzeller-Mischling, Border_Collie, Bulldogge, Cockerspaniel, Dackel, Dobermann, Dogge, Familienhund, Golden_Retriever, Hündchen, Hündin, Jack_Russell, Kätzchen, Labrador_Retriever, Labradore, Leika, Lein_führen, Leonberger, Mischling, Mischlingshund, Mischlingshündin, Pudel, Rehpinscher, Retriever, Riesenschnauzer, Rüde, Schäferhündin, Spaniel, Terriers, Welpe, angeleint, belgisch_Schäferhund, deutsch_Schäferhund, totbeißen*

Search for run 2:

- **Run 2, Iteration 1**:
  - **Terms for the starting point**: *Schäferhund, Labrador*
  - From top 50 nearest neighbors to the starting point:
    - **New candidates**: *Hund, Hündin, Golden_Retriever, Rottweiler, Dackel, Terrier, Dobermann, Vierbeiner, Pitbull, Jack_Russell, Bulldogge, Kampfhund, Riesenschnauzer, angeleint, Bullterrier, Pitbulls, Familienhund, Border_Collie, American_Staffordshire, Labradore, belgisch_Schäferhund, Mischling, Mischlingshund, Staffordshire_Terrier, deutsch_Schäferhund, Pudel, Labrador_Retriever, Hündchen, totbeißen, Hundehalter, Retriever, Dogge, Leonberger, Spaniel, Tierheim, Pitbull-Terrier, Welpe, Rehpinscher, Lein_führen, Rüde, Cockerspaniel, Terriers*
    - **Matches with lexicon**: *Collie, Berner_Sennenhund*
    - **In exclusion list**: *Herrchen, Katze, Frauchen, Büsi, Halterin, Halter*

- **Run 2, Iteration 2**:
  - **Terms for the starting point**: *Collie, Hund*
  - From top 50 nearest neighbors to the starting point:
    - **New candidates**: *Hündin, Vierbeiner, Golden_Retriever, Rottweiler, Border_Collie, Dackel, Jack_Russell, Hundebesitzer, Tier, Hundehalter, Welpe, Haustier, belgisch_Schäferhund, Familienhund, Rehpinscher, anleinen, Pitbull, Leonberger, Labradore, müssen_DogStar, Dobermann, Riesenschnauzer, Retriever, Hauskatze, Labrador_Retriever, Kampfhund, angeleint, Terrier, Meerschweinchen, Tierheim, Pfötli, Mischlingshund, Sennenhunde, Bullterrier, Pony, Rüde, Hündchen, Staffordshire_Terrier, Kaninchen, Lein_führen, Appenzeller-Mischling, Dalmatiner, Pudel*

  - ∗ **Matches with lexicon**: *Schäferhund, Labrador*
  - ∗ **In exclusion list**: *Katze, Herrchen, Büsi, Frauchen, Halterin*

- **Run 2, Iteration 3**:
  - **Terms for the starting point**: *Labrador, Hündin*
  - From top 50 nearest neighbors to the starting point:
    - ∗ **New candidates**: *Hund, Golden_Retriever, Dackel, Vierbeiner, Jack_Russell, Hündchen, Terrier, Rottweiler, Border_Collie, Welpe, Riesenschnauzer, Mischlingshund, Leika, Familienhund, Rüde, Bulldogge, Pitbull, Appenzeller-Mischling, Schäferhündin, Leonberger, belgisch_Schäferhund, Tierheim, Mischling, Dobermann, Bergamasker, angeleint, Pony, Labradore, Dogge, Spaniel, Pudel, Haustier, Bernhardiner, Mischlingshündin, Hirtenhund, Cockerspaniel, Labradorhündin, Pitbulls, deutsch_Schäferhund, Terriers, deutsch_Dogge*
    - ∗ **Matches with lexicon**: *Schäferhund, Berner_Sennenhund, belgisch_Schäferhündin, Collie*
    - ∗ **In exclusion list**: *Herrchen, Frauchen, Katze, Büsi, Halterin*

Assessment of candidates from run 2

- **Candidates (with counts)**: *Golden_Retriever(3), Rottweiler(3), Dackel(3), Terrier(3), Dobermann(3), Vierbeiner(3), Pitbull(3), Jack_Russell(3), Riesenschnauzer(3), angeleint(3), Familienhund(3), Border_Collie(3), Labradore(3), belgisch_Schäferhund(3), Mischlingshund(3), Pudel(3), Hündchen(3), Leonberger(3), Tierheim(3), Welpe(3), Rüde(3), Hund(2), Hündin(2), Bulldogge(2), Kampfhund(2), Bullterrier(2), Pitbulls(2), Mischling(2), Staffordshire_Terrier(2), deutsch_Schäferhund(2), Labrador_Retriever(2), Hundehalter(2), Retriever(2), Dogge(2), Spaniel(2), Rehpinscher(2), Lein_führen(2), Cockerspaniel(2), Terriers(2), Haustier(2), Pony(2), Appenzeller-Mischling(2), American_Staffordshire(1), totbeißen(1), Pitbull-Terrier(1), Hundebesitzer(1), Tier(1), anleinen(1), müssen_DogStar(1), Hauskatze(1)*
- **Added to lexicon**: *Border_Collie, Bulldogge, Cockerspaniel, Dogge, Familienhund, Hündin, Labrador_Retriever, Labradore, Leonberger, Mischling, Rehpinscher, Retriever, Riesenschnauzer, Rüde, Spaniel, belgisch_Schäferhund, deutsch_Schäferhund, müssen_DogStar*

- **Added to shadow lexicon**: *American_Staffordshire, Appenzeller-Mischling, Bullterrier, Dackel, Dobermann, Golden_Retriever, Haustier, Hund, Hundebesitzer, Hundehalter, Hündchen, Jack_Russell, Kampfhund, Lein_führen, Mischlingshund, Pitbull, Pitbull-Terrier, Pitbulls, Pony, Pudel, Rottweiler, Staffordshire_Terrier, Terrier, Terriers, Tierheim, Vierbeiner, Welpe, angeleint, anleinen, totbeißen*

Status before run 3:

- **Updated Lexicon**: *Berner_Sennenhund, Border_Collie, Bulldogge, Chihuahua, Chow_Chow, Cockerspaniel, Collie, Dogge, Familienhund, Hündin, Labrador, Labrador_Retriever, Labradore, Leonberger, Mischling, Rehpinscher, Retriever, Riesenschnauzer, Rüde, Schäferhund, Setter, Spaniel, belgisch_Schäferhund, belgisch_Schäferhündin, deutsch_Schäferhund, müssen_DogStar*
- **Updated Shadow Lexicon**: *American_Staffordshire, Appenzeller-Mischling, Bullterrier, Dackel, Dobermann, Golden_Retriever, Haustier, Hund, Hundebesitzer, Hundehalter, Hündchen, Jack_Russell, Kampfhund, Kätzchen, Leika, Lein_führen, Mischlingshund, Mischlingshündin, Pitbull, Pitbull-Terrier, Pitbulls, Pony, Pudel, Rottweiler, Schäferhündin, Staffordshire_Terrier, Terrier, Terriers, Tierheim, Vierbeiner, Welpe, angeleint, anleinen, totbeißen*

Search for run 3:

- **Run 3, Iteration 1**:
  - **Terms for the starting point**: *Schäferhund, Labrador*
  - From top 50 nearest neighbors to the starting point:
    - **New candidates**: *Hund, Golden_Retriever, Rottweiler, Dackel, Terrier, Dobermann, Vierbeiner, Pitbull, Jack_Russell, Kampfhund, angeleint, Bullterrier, Pitbulls, American_Staffordshire, Mischlingshund, Staffordshire_Terrier, Pudel, Hündchen, totbeißen, Hundehalter, Tierheim, Pitbull-Terrier, Welpe, Lein_führen, Terriers*
    - **Matches with lexicon**: *Hündin, Bulldogge, Riesenschnauzer, Familienhund, Collie, Border_Collie, Labradore, belgisch_Schäferhund, Mischling, deutsch_Schäferhund, Labrador_Retriever, Berner_Sennenhund, Retriever, Dogge, Leonberger, Spaniel, Rehpinscher, Rüde, Cockerspaniel*
    - **In exclusion list**: *Herrchen, Katze, Frauchen, Büsi, Halterin, Halter*

- **Run 3, Iteration 2**:
  - **Terms for the starting point**: *Hündin, Golden_Retriever*
  - From top 50 nearest neighbors to the starting point:
    - **New candidates**: *Hund, Rottweiler, Jack_Russell, Vierbeiner, Dackel, Terrier, Hündchen, Pitbull, Mischlingshund, Welpe, Pitbulls, American_Staffordshire, Leika, angeleint, Bullterrier, Staffordshire_Terrier, Dobermann, Zwergpinscher, Schäferhündin, Appenzeller-Mischling, bei_Gassigehen, Labradorhündin, Pudel, Haustier, Tierheim, Kampfhund, Mischlingshündin, Pony, Schäfermischling*
    - **Matches with lexicon**: *Schäferhund, Labrador, Border_Collie, Riesenschnauzer, Leonberger, Rüde, Familienhund, Berner_Sennenhund, Bulldogge, belgisch_Schäferhündin, belgisch_Schäferhund, Labradore, Rehpinscher, Cockerspaniel, Collie, müssen_DogStar*
    - **In exclusion list**: *Katze, Herrchen, Büsi, Frauchen, Halterin*

- **Run 3, Iteration 3**:
  - **Terms for the starting point**: *Schäferhund, Rottweiler*
  - From top 50 nearest neighbors to the starting point:
    - **New candidates**: *Hund, Kampfhund, Pitbull, Dobermann, Pitbulls, Golden_Retriever, Bullterrier, Terrier, American_Staffordshire, Dackel, Pitbull-Terrier, Staffordshire_Terrier, totbeißen, American_Staffordshire-Terrier, Tod_beißen, angeleint, Vierbeiner, Hundehalter, Hundebesitzer, Mastiff, Dogo_Argentino, Jack_Russell, Maulkorb_tragen, einschläfern, Staffordshire_Bullterrier, Hunderasse, Wesenstest, Lein_führen, anleinen, American_Pitbull, Pudel*
    - **Matches with lexicon**: *Hündin, Riesenschnauzer, Familienhund, Bulldogge, Labrador, Leonberger, deutsch_Schäferhund, Labradore, Border_Collie, Dogge, Rehpinscher, Collie, Cockerspaniel*
    - **In exclusion list**: *Herrchen, Frauchen, Katze, Büsi, Halterin*

Assessment of candidates from run 3

- **Candidates (with counts)**: *Hund(3), Dackel(3), Terrier(3), Dobermann(3), Vierbeiner(3), Pitbull(3), Jack_Russell(3), Kampfhund(3), angeleint(3), Bullterrier(3), Pitbulls(3), American_Staffordshire(3), Staffordshire_Terrier(3), Pudel(3), Golden_Retriever(2), Rottweiler(2), Mischlingshund(2), Hündchen(2), totbeißen(2), Hundehalter(2), Tierheim(2), Pitbull-Terrier(2), Welpe(2), Lein_führen(2), Terriers(1), Leika(1), Zwergpinscher(1), Schäferhündin(1), Appenzeller-Mischling(1), bei_Gassigehen(1), Labradorhündin(1), Haustier(1), Mischlingshündin(1), Pony(1), Schäfermischling(1), American_Staffordshire-Terrier(1), Tod_beißen(1), Hundebesitzer(1), Mastiff(1), Dogo_Argentino(1), Maulkorb_tragen(1), einschläfern(1), Staffordshire_Bullterrier(1), Hunderasse(1), Wesenstest(1), anleinen(1), American_Pitbull(1)*

- **Added to lexicon**: *American_Pitbull, American_Staffordshire, Appenzeller-Mischling, Dackel, Dobermann, Dogo_Argentino, Golden_Retriever, Hund, Jack_Russell, Labradorhündin, Lein_führen, Mastiff, Maulkorb_tragen, Mischlingshund, Pitbull, Pitbull-Terrier, Pudel, Rottweiler, Schäferhündin, Staffordshire_Bullterrier, Terrier, Terriers, Zwergpinscher, angeleint, anleinen, bei_Gassigehen*

- **Added to shadow lexicon**: *American_Staffordshire-Terrier, Bullterrier, Haustier, Hundebesitzer, Hundehalter, Hunderasse, Hündchen, Kampfhund, Leika, Mischlingshündin, Pitbulls, Pony, Schäfermischling, Staffordshire_Terrier, Tierheim, Tod_beißen, Vierbeiner, Welpe, Wesenstest, einschläfern, totbeißen*

Finally, the lexicon contains the following terms after the three runs:
American_Pitbull, American_Staffordshire, Appenzeller-Mischling, Berner_Sennenhund, Border_Collie, Bulldogge, Chihuahua, Chow_Chow, Cockerspaniel, Collie, Dackel, Dobermann, Dogge, Dogo_Argentino, Familienhund, Golden_Retriever, Hund, Hündin, Jack_Russell, Labrador, Labrador_Retriever, Labradore, Labradorhündin, Lein_führen, Leonberger, Mastiff, Maulkorb_tragen, Mischling, Mischlingshund, Pitbull, Pitbull-Terrier, Pudel, Rehpinscher, Retriever, Riesenschnauzer, Rottweiler, Rüde, Schäferhund, Schäferhündin, Setter, Spaniel, Staffordshire_Bullterrier, Terrier, Terriers, Zwergpinscher, angeleint, anleinen, bei_Gassigehen, belgisch_Schäferhund, belgisch_Schäferhündin, deutsch_Schäferhund, müssen_DogStar

When we examine the resulting lexicon (52 entries), we make several observations. First of all, most of the new entries are indeed dog breeds as desired (41 entries, 79%). There are some terms which are only variations of others (i.e., female forms as *Labradorhündin, belgisch_Schäferhündin* and wrongly lemmatized word forms like the plural of *Labrador*, i.e., *Labradore*, or the genitive of *Terrier*, i.e., *Terriers*).

There are also two versions for a crossbreed (*Mischling, Mischlingshund*) which is not a dog breed per se, but certainly *crossbreed* is an acceptable instance regarding semantic similarity. Accepting or deleting such entries lies finally in the responsibility of the user.

Four entries refer to generic designations of dogs (*Familienhund, Hund*) or to the more specific male or female designations of a dog (*Hündin, Rüde*). These are hypernyms or designators of dogs stemming from a more generic level.

Furthermore, there are also five terms which are clearly semantically related to dog breeds—in the sense that they describe actions correlated with having a dog—but are certainly to filter out in a dog breed lexicon: *Lein_führen (keeping the dog on a leash), Maulkorb_tragen (to wear a muzzle), angeleint, anleinen (put on a leash), bei_Gassigehen (taking the dog for a walk).*

Lastly, *müssen_DogStar* is clearly an error which is also a falsely created collocation by the standard techniques of `word2vec`.

If we keep in mind the mixed types of similarities that are present and in a sense fused in the embedding model, it is a mere consequence that for example hypernyms or activities that are strongly correlated with the target show up in the induction process.

On the one hand, we attempt to filter them out, either explicitly through the exclusion lexicon, or implicitly by the assessment procedure which relies heavily on the given initial lexicon. On the other hand, we try to make use of this additional information by collecting such terms in the shadow lexicon so that it serves as a proxy for the assessment of good candidates. This leads in turn also to the case where the shadow lexicon occasionally contributes to erroneously increase the scores (based on similarity) of further similar but undesired terms during the calculations in the assessment step. But fortunately, this contribution tends to be mitigated by the fact that numerous ways exist that mislead the search, but they are—in general—different from each other and therefore not self-reinforcing.

If we put the induction procedure as such and its intermediate results under investigation, we might stumble upon the fact that after the first run, in the first assessment step, we only add *belgisch_Schäferhündin* to the lexicon. Note that this is the 49^{th} candidate out of 50, ordered by the frequency of occurrences in the three iterations of the first run. This seems odd given that we have many candidates which appear to be more obvious or probable at least based on the fact that they showed up in each iteration of the first run, i.e., three times (*Hund, Rottweiler, Dackel, Terrier, Vierbeiner, Pitbull, Jack_Russell, Riesenschnauzer, Familienhund, Border_Collie, Labradore, belgisch_Schäferhund, Mischlingshund, deutsch_Schäferhund, Pudel, Hündchen, Leonberger, Rüde, Cockerspaniel*).

The reason why only *belgisch_Schäferhündin* crossed the barrier to enter the lexicon is linked to the properties of the shadow lexicon which is populated at run-time. In this case, this refers to the fact that none of the 48 candidates which were algorithmically assessed before *belgisch_Schäferhündin* met the threshold to be added to the lexicon. But many of them were added to the shadow lexicon. By increasing the contribution of the terms from the shadow lexicon with each additional entry, the collective evidentiary power—including the evidence from the comparison with the small given initial lexicon—was enough for the 49^{th} candidate, *belgisch_Schäferhündin*, to meet the threshold so that we add the term to the lexicon. This demonstrates nicely again how important the contribution of aggregated gray (or unsure) information in the shadow lexicon is for the lexicon induction procedure.

The goal of this introductory example was not to present a perfect showcase, but rather to elucidate how the induction process works considering its two main parts of search and assessment of candidates. It also shows that the completely automated filtering of false entries from the generated candidates is a challenging task. The terms that are related to dogs but are not dog breeds themselves are such ones which we strive to keep in the shadow lexicon. They are not to be included in the lexicon but nevertheless they contribute to identify new good candidates that should be accepted as new entries. This contribution during the assessment phase is an important element of the lexicon induction process as we also see in the number of added new entries after run 2 and 3 when we have a populated shadow lexicon in contrast to the single new entry from the run 1.

However, the example also suggests that a long chain of consecutive runs without clearing the shadow lexicon may lead to an increasing number of false positives. The five terms which are only related to having a dog are all included after run 3, while after run 2 only the generic *Rüde* and the erroneous collocation *müssen_DogStar* are wrongly added.

In the next section, we will now turn to a more linguistically and thematically related case study where we explore the possibility of steering the induction process. For the sake of brevity and clarity we report the results in a less extensive format.

6.2 Injected Guidance for Lexicon Induction

Sometimes, the main purpose for a lexical resource is to cover a specific phenomenon which is a subset of a more common one. For example, let us suppose we would be interested in the detection of specific ways of communication, and further we would be interested in communication that is coupled with negative sentiment.

To be clearer, we would like to detect not only locations in texts where people *say*, *speak*, or *talk* but more text positions where people *rant*, *complain*, or *grouse*. To create a resource which aims at the detection of such instances of communication, it would be helpful if we could just define the desired combination of concepts (i.e., communication and negative sentiment).

We have claimed that one of the strengths of the lexicon induction process is its versatility, and, as one part of this, the possibility to adapt or shape the result in the desired way. We report in this section on two exemplary cases to illustrate this possibility. Firstly, to come back to the example provided beforehand, we detect a set of verbs of communication which also convey a negative sentiment. Secondly, we induce a lexicon which captures domain-specific crime terms (from the financial sector).

6.2.1 Communication with Negative Sentiment

For the first case, we start with a small set of general verbs of communication and "inject" our desired steering momentum through a given starting point. Trivially, we start with some of the most common verbs of communication: *sagen (to say)*, *sprechen (to speak)*, and *reden (to talk)*. We then combine them with a set of terms (i.e., negative emotions) to get a mixture of these two concepts (verbs of communication and negative sentiment). More precisely, we add *Wut (rage)*, *Ärger (anger)*, and *Zorn (fury)*.

10 most similar entries to "sagen", "reden", "sprechen", "Wut", "Ärger" and "Zorn"		
Rank	Word	Similarity
1	Frustration	0.6302
2	Frust	0.6183
3	Empörung	0.6140
4	Unmut	0.6117
5	schimpfen	0.6070
6	klagen	0.6034
7	ärgern	0.5898
8	Enttäuschung	0.5844
9	beklagen	0.5733
10	Angst	0.5713

TABLE 6.1: 10 most similar terms to *sagen*, *reden*, *sprechen*, *Wut*, *Ärger*, and *Zorn* in the semantic space of the word2vec model, ordered by cosine similarity

Table 6.1 shows the result of a simple combination of the two concepts. Note that this result is different from other combinations of terms (e.g. Table 5.8) in the sense that the given terms are excluded from the result.

We take now the verbs of this result, which are *schimpfen (to rant)*, *klagen (to lament)*, *ärgern (to huff)*, and *beklagen (to complain)* and add them to our seed lexicon of verbs

of communication. This already leads to a drift towards verbs of "negative communication". Furthermore, we keep the negative emotions as starting point to guide the induction process. We then run the process for three recurrent runs with ten iterations each (with default parameters).

Search for "negative communication"; Status before start (run 1):

- **Lexicon**: *sagen, reden, sprechen, schimpfen, klagen, ärgern, beklagen*
- **Shadow Lexicon**: *[]*
- **Exclusion Lexicon**[5]: *lachen, amüsieren, scherzen, lächeln, frohlocken, grün_Klee, schmunzeln, kichern, verschmitzt, plaudern, gut_gelaunt, flachsen, strahlen, grüssen, umarmen, freuen, Lächeln, Grinsen, zurücklächeln_zurück, jauchzen, glucksen, grinsen*

We see in Table 6.2 that we were at least partly successful in blending both concepts when we inspect the new terms. We have added 19 terms to the lexicon (*aufregen, befremden, beschweren, beunruhigt, echauffieren, empören, enervieren, entrüsten, entrüstet, entsetzen, ereifern, erschrecken, herziehen, irritiert, jammern, lamentieren, lästern, mokieren, sehr_enttäuscht*). Although *entrüstet (outraged)*, *irritiert (irritated)*, and *sehr_enttäuscht (very disappointed)* are not verbs in the infinitive form, they represent the status of having a negative sentiment. These forms are syntagmatically similar enough to the verbs we are looking for so that they are represented in the close vicinity in the embedding model. Since the induction process is geared to deal with uncertainty and to be tolerant against terms in the lexicon which do not fully match the desired concept properties (remember that *sagen, sprechen, reden* also just satisfy the criterion of communication but not the one of of negative sentiment), we will leave these terms in the lexicon for further induction[6].

As we have mentioned earlier in Section 4.3, it is generally more productive to change the steering during the induction process than to use the same starting point over and over again for sequential induction runs. In other words, we will change the starting point (the set of given terms in the beginning of the search) for the next search round. This leads us to the task of choosing such new terms. The idea that we apply here is

[5]These terms were identified as introducing noise in a preliminary test. Although not encompassing, they represent mainly verbs (partly verbs of communication) coupled with positive sentiment.

[6]Of course, the user may withdraw any number of entries from the lexicon during the induction process, and, additionally may even put them into the exclusion lexicon for further runs.

Re-Sampling during Search		
Run	Iteration	Starting Point Terms
1	1	*Ärger, Wut, Zorn*
1	2	schimpfen,Frustration
1	3	ärgern,Wut
1	4	klagen,Ärger
1	5	beklagen,Unmut
1	6	schimpfen,beschweren
1	7	reden,empören
1	8	klagen,empört
1	9	ärgern,empören
1	10	schimpfen,ereifern
2	1	*Ärger, Wut, Zorn*
2	2	schimpfen,Frustration
2	3	ärgern,Wut
2	4	klagen,Ärger
2	5	beklagen,Unmut
2	6	schimpfen,beschweren
2	7	reden,empören
2	8	klagen,empört
2	9	ärgern,empören
2	10	schimpfen,ereifern
3	1	*Ärger, Wut, Zorn*
3	2	schimpfen,Frustration
3	3	ärgern,Wut
3	4	klagen,Ärger
3	5	beklagen,Unmut
3	6	schimpfen,beschweren
3	7	reden,empören
3	8	klagen,empört
3	9	ärgern,empören
3	10	schimpfen,ereifern

Lexicon Content	
Shadow Lexicon after 3 Runs	Lexicon after 3 Runs
Kopf_schütteln, abfällig, abschätzig, bedauern, bekräftigen, beschwichtigen, beschämen, besorgt, bestürzt, beteuern, betonen, beunruhigen, darüber_aufregen, einräumen, empört, entsetzt, enttäuschen, enttäuscht, erbosen, erbost, erfreut, erklären, erschreckt, erstaunt, erzürnen, erzürnt, frustrieren, frustriert, genervt, geärgert, herzeihen, höhnen, irritieren, konsternieren, mangelnd_Unterstützung, meckern, motzen, nennen, nerven, perplex, poltern, präzisieren, relativieren, schockieren, schockiert, schreiben, spotten, ungehalten, unzufrieden, versichern, verwundert, verärgern, verärgert, wettern, witzeln, wundern, wütend, zetern, überrascht	aufregen, befremden, beklagen, beschweren, beunruhigt, echauffieren, empören, enervieren, entrüsten, entrüstet, entsetzen, ereifern, erschrecken, herziehen, irritiert, jammern, klagen, lamentieren, lästern, mokieren, reden, sagen, schimpfen, sehr_enttäuscht, sprechen, ärgern

TABLE 6.2: Re-sampling of terms for the starting point for each iteration during search and content of the lexicons after three recurrent runs (with the initial starting point *Ärger, Wut, Zorn*) without flushing the shadow lexicon.

that we would benefit from the additional information, i.e., the new terms that we have already found, and use the lexicon in its entirety to estimate the best new candidates for the new starting point. In order to do so, we will now use an intermediate step to consolidate the results that we already have in our lexicon (26 terms; see Table 6.2).

In this intermediate step, we cluster the current lexicon into 10 clusters, using the same method as described in Chapter 5 to re-embed the created lexicon in the semantic space. As a next step, we take the centroid of each cluster and find the three closest terms in the embedding. These three terms are considered as candidate triples from which we will select the most promising ones.

Table 6.3 shows the three closest terms for the centroids of the cluster model. These three terms in turn are used to form the basis for a new query for the 30 most similar terms (i.e., we actually take the mean of the three terms and use this point to look for the nearest neighbors.[7])

[7]This point is subtly different from the original centroid. In focusing on the top three nearest neighbors from the centroid, the point will be just be marginally closer to this three terms. However, the main reason to use this point instead of the original centroid is rather that we use the three terms as starting point for the next search in the lexical induction.

Resulting cluster model for further selection of starting point candidate triples		
Cluster/Centroid Number	3 Most Similar Terms to Centroid	Terms in Lexicon Relative to Combination
1	herziehen, herzeihen, lästern	11
2	ärgern, ereifern, aufregen	15
3	entsetzen, empört, empören	8
4	reden, sprechen, meinen	1
5	sagen, erklären, meinen	1
6	lamentieren, jammern, schimpfen	12
7	lästern, schimpfen, mokieren	13
8	irritiert, befremden, erstaunt	5
9	klagen, beklagen, beschweren	11
10	beunruhigt, beunruhigen, besorgt	6

TABLE 6.3: Three most similar terms to the centroids of the cluster model (10 clusters) calculated with the lexicon (26 terms) after three runs, and estimated aptness for starting point candidate triple, given by known terms in the lexicon

When we compare the 30 nearest neighbors to the mean of the candidate triple with the given 26 term lexicon, we are able to estimate the aptness of the candidate triple as a starting point for the next run in the induction process.

We observe that two candidate triples (*reden, sprechen, meinen* and *sagen, erklären, meinen*) each have only one term regarding their 30 nearest neighbors which is already part of the lexicon. Those candidate triples are stemming from the clusters which represented the original "generic" verbs of communication we used in the first place to initialize the lexicon (*sagen, reden, sprechen*) and are therefore no good candidate triples for a new starting point in the induction process.

The highest scoring candidate triples are *ärgern, ereifern, aufregen* (15 known terms), *lästern, schimpfen, mokieren* (13 known terms), *lamentieren, jammern, schimpfen* (12 known terms), and *klagen, beklagen, beschweren* (11 known terms) as well as *herziehen, herzeihen*[8]*, lästern* (11 known terms).

Since *schimpfen* (as one of the originally added verbs of communication with negative sentiment) appears in two candidate triples, we firstly start the next run with the starting point formed by the triple *lästern, schimpfen, mokieren.* This time, we perform again

So we also estimate the aptness of the combination of these three terms. One way to do this is to compare the resulting set with the known lexicon to see how many of the candidates will be known or unknown. While this is actually a simulation of the first step of the lexical induction process, it helps to filter out new starting points which were found by the clustering but which do not represent a meaningful subconcept of the lexicon.

[8]This is an error from the lemmatization process. Note that the semantics of the erroneous lemma is coherent in the sense that it is represented closely (cosine similarity of 0.70) to the correct lemma *herziehen*

three consecutive runs and inspect the result right afterwards. Additionally, we clear the shadow lexicon to follow the search path of the new starting point more directly[9].

Status before start (run 2):

- **Lexicon**: *aufregen, befremden, beklagen, beschweren, beunruhigt, echauffieren, empören, enervieren, entrüsten, entrüstet, entsetzen, ereifern, erschrecken, herziehen, irritiert, jammern, klagen, lamentieren, lästern, mokieren, reden, sagen, schimpfen, sehr_enttäuscht, sprechen, ärgern*
- **Shadow Lexicon**: *[]*
- **Exclusion Lexicon**: *lachen, amüsieren, scherzen, lächeln, frohlocken, grün_Klee, schmunzeln, kichern, verschmitzt, plaudern, gut_gelaunt, flachsen, strahlen, grüssen, umarmen, freuen, Lächeln, Grinsen, zurücklächeln_zurück, jauchzen, glucksen, grinsen*

Table 6.4 shows us that we have easily doubled the lexicon size and now have 56 terms that correspond fairly closely to the intended merge of concepts (communication and negative sentiment). The new terms comprise of *Gejammer, abfällig, bitterlich, empört, erbosen, frotzelen, gewitzelt, geärgert, giftelt, giftelte, herzeihen, höhnen, klönen, maulen, meckern, motzen, nerven, nörgeln, schnödet, schwadronieren, spotten, spötteln, sticheln, stänkern, ungehalten, verärgert, wettern, wundern, wütend*, and *zetern.*

We will carry out a last round of induction, using the candidate triple *klagen, beklagen, beschweren*[10]. Since the shadow lexicon also contains many good candidates, we additionally merge the lexicon (56 terms) with the shadow lexicon (67 terms) before we start the next run.

[9] One could also keep the shadow lexicon in this case because it is rather small. But where the shadow lexicon allows to make use of uncertain information, it also increases the probability of false-positives and therefore it is a good advice to let it grow for new promising search paths independently, i.e. to flush it from other searches performed beforehand.

[10] We choose here the triple *klagen, beklagen, beschweren* instead of the triple *ärgern, ereifern, aufregen* which scored even higher on the intermediate clustering step considering the aptness estimation. We decide to take the candidate which contains two of the originally added verbs of negative sentiment (*klagen, beklagen*). However, although the result for the triple *ärgern, ereifern, aufregen* as starting point would be different, both choices represent the original intention well for this illustrative example which is not geared towards optimization anyway.

Re-Sampling during Search		
Run	Iteration	Starting Point Terms
1	1	*lästern, schimpfen, mokieren*
1	2	jammern, wettern
1	3	lamentieren', poltern
1	4	jammern, wettern
1	5	lamentieren, poltern
1	6	jammern, zetern
1	7	klagen, stöhnen
1	8	jammern, wettern
1	9	lamentieren, Gejammer
1	10	jammern, Jammern
2	1	*lästern, schimpfen, mokieren*
2	2	lamentieren, spotten
2	3	mokieren, wettern
2	4	schimpfen, spotten
2	5	lästern, wettern
2	6	lamentieren, spotten
2	7	jammern, wettern
2	8	schimpfen, poltern
2	9	empören, fluchen
2	10	aufregen, wütend
3	1	*lästern, schimpfen, mokieren*
3	2	spotten, poltern
3	3	wettern, witzeln
3	4	spotten, frotzelen
3	5	höhnen, witzeln
3	6	spötteln, frotzelen
3	7	höhnen, witzeln
3	8	schimpfen, zurufen
3	9	jammern, fluchen
3	10	motzen, stöhnen

Lexicon Content	
Shadow Lexicon after 3 Runs	Lexicon after 3 Runs
Gerede, Jammerei, Jammern, Klagelied, Kopf_schütteln, Lamentieren, Lamento, Leviten_lesen, Meckern, Nörgeln, Rage, Rohrspatz, Wehklage, Wütend, abschätzig, anprangern, aufschreien, beklagt, belehren, beschämen, bös_Wort, deprimieren, endlich_aufhören, entsetzt, enttäuscht, erbost, erstaunt, erzürnen, fassungslos, feixen, fluchen, frustrieren, geifern, geisseln, genervt, grollen, grübeln, irritieren, kalauern, keifen, lauthals, murren, orakeln, palavern, perplex, polemisieren, poltern, raunzen, räsonieren, schockieren, schockiert, schämen, sinnieren, sprachlos, staunen, stöhnen, tuscheln, unken, verstören, verächtlich, verärgern, vorhalten, wehklagen, witzeln, witzelt, zurechtweisen, ätzen	Gejammer, abfällig, aufregen, befremden, beklagen, beschweren, beunruhigt, bitterlich, echauffieren, empören, empört, enervieren, entrüsten, entrüstet, entsetzen, erbosen, ereifern, erschrecken, frotzelen, gewitzelt, geärgert, giftelt, giftelte, herzeihen, herziehen, höhnen, irritiert, jammern, klagen, klönen, lamentieren, lästern, maulen, meckern, mokieren, motzen, nerven, nörgeln, reden, sagen, schimpfen, schnödet, schwadronieren, sehr_enttäuscht, spotten, sprechen, spötteln, sticheln, stänkern, ungehalten, verärgert, wettern, wundern, wütend, zetern, ärgern

TABLE 6.4: Re-sampling of terms for the starting point for each iteration during search and content of the lexicons after three additional recurrent runs without flushing the shadow lexicon. Starting point is defined by the triple *lästern, schimpfen, mokieren*

STATUS BEFORE START (RUN 3):

- **Lexicon**: *Gejammer, abfällig, aufregen, befremden, beklagen, beschweren, beunruhigt, bitterlich, echauffieren, empören, empört, enervieren, entrüsten, entrüstet, entsetzen, erbosen, ereifern, erschrecken, frotzelen, gewitzelt, geärgert, giftelt, giftelte, herzeihen, herziehen, höhnen, irritiert, jammern, klagen, klönen, lamentieren, lästern, maulen, meckern, mokieren, motzen, nerven, nörgeln, reden, sagen, schimpfen, schnödet, schwadronieren, sehr_enttäuscht, spotten, sprechen, spötteln, sticheln, stänkern, ungehalten, verärgert, wettern, wundern, wütend, zetern, ärgern, Gerede, Jammerei, Jammern, Klagelied, Kopf_schütteln, Lamentieren, Lamento, Leviten_lesen, Meckern, Nörgeln, Rage, Rohrspatz, Wehklage, Wütend, abschätzig, anprangern, aufschreien, beklagt, belehren, beschämen, bös_Wort, deprimieren, endlich_aufhören, entsetzt, enttäuscht, erbost, erstaunt, erzürnen, fassungslos, feixen, fluchen, frustrieren, geifern, geisseln, genervt, grollen, grübeln,*

irritieren, kalauern, keifen, lauthals, murren, orakeln, palavern, perplex, polemisieren, poltern, raunzen, räsonieren, schockieren, schockiert, schämen, sinnieren, sprachlos, staunen, stöhnen, tuscheln, unken, verstören, verächtlich, verärgern, vorhalten, wehklagen, witzeln, witzelt, zurechtweisen, ätzen

- **Shadow Lexicon**: *[]*
- **Exclusion Lexicon**: *lachen, amüsieren, scherzen, lächeln, frohlocken, grün_Klee, schmunzeln, kichern, verschmitzt, plaudern, gut_gelaunt, flachsen, strahlen, grüssen, umarmen, freuen, Lächeln, Grinsen, zurücklächeln_zurück, jauchzen, glucksen, grinsen*

Table 6.5 represents the state of the induction after the three runs with the triple *klagen, beklagen, beschweren* as a starting point. We have now 144 terms in the lexicon, 21 of them are newly induced. The new terms are: *Kritik_üben, ankreiden, bemängeln, bemängelt, besorgt, beunruhigen, darüber_beschweren, empört_darüber, harsch_kritisieren, heftig_kritisieren, konsternieren, kritisieren, kritisiert, missfallen, monieren, moniert, rügen, sauer_aufstoßen, scharf_kritisieren, vorwerfen*, and *überrascht*.

Up to this point, this procedure has yielded a result which consists of over 100 verbs that convey a negative sentiment. Since we do not provide any extrinsic evaluation for this illustrative example, we nevertheless provide a second view on the result besides the bare word lists given in Table 6.5.

Again, we re-embed the lexicon (including this time also the shadow lexicon but excluding the initially used verbs of communication *reden, sagen, sprechen*) and cluster into 10 clusters. We then look at the nearest neighbors of the centroids (see Table 6.6) which nicely generalize the different sub-concepts in our lexicon.

Re-Sampling during Search		
Run	Iteration	Starting Point Terms
1	1	*klagen, beklagen, beschweren*
1	2	empören, kritisieren
1	3	beklagen, monieren
1	4	empören, bemängeln
1	5	beklagen, monieren
1	6	ärgern, kritisieren
1	7	klagen, monieren
1	8	ärgern, bemängeln
1	9	beklagen, kritisieren
1	10	empören, monieren
2	1	*klagen, beklagen, beschweren*
2	2	schimpfen, kritisieren
2	3	klagen, monieren
2	4	beschweren, kritisieren
2	5	empören, monieren
2	6	ereifern, bemängeln
2	7	beschweren, monieren
2	8	ärgern, kritisieren
2	9	verärgern, bemängeln
2	10	beklagen, monieren
3	1	*klagen, beklagen, beschweren*
3	2	empören, wehren
3	3	ärgern, protestieren
3	4	klagen, wehren
3	5	beschweren, Klage
3	6	klagen, Beschwerde
3	7	beklagen, Klage
3	8	beschweren, Beschwerde
3	9	klagen, Beschwerde_eingereichen
3	10	beschweren, Klage

Lexicon Content	
Shadow Lexicon after 3 Runs	Lexicon after 3 Runs
Dagegen_wehren, Kritik, Kritiker, Vorwurf, Vorwurf_erheben, argumentieren, argwöhnen, beanstanden, bedauern, befremdet_darüber, befürchten, behaupten, beteuern, betonen, bezeichnen, bezichtigen, bezweifeln, brandmarken, dagegen_wehren, darauf_hinweisen, darüber_beklagen, davor_warnen, einräumen, entgegnen, enttäuschen, ergehen_lassen, erstaunen, erstaunt_darüber, festhalten_fest, feststellen_fest, fordern, fürchten, harsch_Kritik, konstatieren, konteren, meinen, mutmassen, reagieren_empört, reklamieren, rüffeln, scharf_Kritik, skeptisch, stören, tadeln, unverständlich, unzufrieden, urteilt, verunsichern, verweisen_darauf, verärgert_darüber, wandte, warnen, warnen_davor, widersprechen, Ärger, ärgerlich, äussern, üben_Kritik, überraschen	Gejammer, Gerede, Jammerei, Jammern, Klagelied, Kopf_schütteln, Kritik_üben, Lamentieren, Lamento, Leviten_lesen, Meckern, Nörgeln, Rage, Rohrspatz, Wehklage, Wütend, abfällig, abschätzig, ankreiden, anprangern, aufregen, aufschreien, befremden, beklagen, beklagt, belehren, bemängeln, bemängelt, beschweren, beschämen, besorgt, beunruhigen, beunruhigt, bitterlich, bös_Wort, darüber_beschweren, deprimieren, echauffieren, empören, empört, empört_darüber, endlich_aufhören, enervieren, entrüsten, entrüstet, entsetzen, entsetzt, enttäuscht, erbosen, erbost, ereifern, erschrecken, erstaunt, erzürnen, fassungslos, feixen, fluchen, frotzelen, frustrieren, geifern, geisseln, genervt, gewitzelt, geärgert, giftelt, giftelte, grollen, grübeln, harsch_kritisieren, heftig_kritisieren, herzeihen, herziehen, höhnen, irritieren, irritiert, jammern, kalauern, keifen, klagen, klönen, konsternieren, kritisieren, kritisiert, lamentieren, lauthals, lästern, maulen, meckern, missfallen, mokieren, monieren, moniert, motzen, murren, nerven, nörgeln, orakeln, palavern, perplex, polemisieren, poltern, raunzen, reden, räsonieren, rügen, sagen, sauer_aufstoßen, scharf_kritisieren, schimpfen, schnödet, schockieren, schockiert, schwadronieren, schämen, sehr_enttäuscht, sinnieren, spotten, sprachlos, sprechen, spötteln, staunen, sticheln, stänkern, stöhnen, tuscheln, ungehalten, unken, verstören, verächtlich, verärgern, verärgert, vorhalten, vorwerfen, wehklagen, wettern, witzeln, witzelt, wundern, wütend, zetern, zurechtweisen, ärgern, ätzen, überrascht

TABLE 6.5: Re-sampling of terms for the starting point for each iteration during search and content of the lexicons after three additional recurrent runs without flushing the shadow lexicon. Starting point is defined by the triple *klagen, beklagen, beschweren*

10 most similars to centroid 1		
Rank	Word	Similarity
1	ärgern	0.8446
2	aufregen	0.8104
3	empören	0.8086
4	ereifern	0.8001
5	wundern	0.7953
6	enervieren	0.7525
7	nerven	0.7521
8	echauffieren	0.7326
9	beschweren	0.7069
10	schimpfen	0.7046

10 most similars to centroid 2		
Rank	Word	Similarity
1	schimpfen	0.8350
2	lästern	0.8153
3	spotten	0.7589
4	wettern	0.7553
5	höhnen	0.7360
6	mokieren	0.7053
7	poltern	0.6922
8	witzeln	0.6870
9	frotzelen	0.6588
10	sticheln	0.6502

10 most similars to centroid 3		
Rank	Word	Similarity
1	monieren	0.8173
2	verweisen_darauf	0.7943
3	feststellen_fest	0.7496
4	bemängeln	0.7485
5	festhalten_fest	0.7443
6	darauf_hinweisen	0.7361
7	erinnern_daran	0.7124
8	bezweifeln	0.6918
9	kritisieren	0.6844
10	warnen_davor	0.6546

10 most similars to centroid 4		
Rank	Word	Similarity
1	verärgern	0.8283
2	erstaunt	0.7965
3	beunruhigen	0.7880
4	irritieren	0.7811
5	enttäuscht	0.7760
6	überrascht	0.7759
7	enttäuschen	0.7619
8	empört	0.7576
9	verärgert	0.7518
10	entsetzen	0.7375

10 most similars to centroid 5		
Rank	Word	Similarity
1	beklagen	0.8139
2	klagen	0.8025
3	beschweren	0.7938
4	ärgern	0.7222
5	monieren	0.6643
6	empören	0.6560
7	kritisieren	0.6453
8	reklamieren	0.6378
9	fürchten	0.6336
10	wehren	0.6311

10 most similars to centroid 6		
Rank	Word	Similarity
1	kritisieren	0.8106
2	betonen	0.8106
3	argumentieren	0.7744
4	meinen	0.7467
5	behaupten	0.7465
6	monieren	0.7316
7	entgegnen	0.7274
8	erklären	0.7154
9	beteuern	0.6980
10	widersprechen	0.6976

10 most similars to centroid 7		
Rank	Word	Similarity
1	entsetzen	0.8458
2	empört	0.8232
3	schockiert	0.8155
4	wütend	0.7822
5	fassungslos	0.7733
6	entsetzt	0.7655
7	schockieren	0.7510
8	entrüstet	0.7464
9	erschrecken	0.7433
10	konsternieren	0.7344

10 most similars to centroid 8		
Rank	Word	Similarity
1	jammern	0.8229
2	schimpfen	0.7601
3	fluchen	0.7345
4	lamentieren	0.7112
5	lästern	0.7004
6	motzen	0.6984
7	Gejammer	0.6977
8	meckern	0.6879
9	nerven	0.6756
10	stöhnen	0.6693

10 most similars to centroid 9		
Rank	Word	Similarity
1	harsch_Kritik	0.8466
2	scharf_Kritik	0.8398
3	heftig_Kritik	0.8102
4	Kritik	0.8042
5	heftig_kritisieren	0.7865
6	kritisieren	0.7629
7	Kritik_üben	0.7456
8	scharf_kritisieren	0.7434
9	kritisiert	0.7379
10	üben_Kritik	0.7359

10 most similars to centroid 10		
Rank	Word	Similarity
1	bezichtigen	0.8106
2	anprangern	0.7827
3	kritisieren	0.7763
4	rügen	0.7754
5	geisseln	0.7740
6	vorwerfen	0.7734
7	scharf_kritisieren	0.7712
8	vorhalten	0.7619
9	tadeln	0.7061
10	monieren	0.7054

TABLE 6.6: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for communication with negative sentiment in the semantic space of the word2vec model, ordered by cosine similarity

6.2.2 Crimes in the Financial Sector

As we have mentioned in the beginning of this section, we will now demonstrate in a second example how we adapt a lexical resource. This time, the main intention is to find terms for crimes in the finance sector. Such a derived list—or more precisely: concepts—could be further used to measure which sorts of crimes were linked to what types of actors, or even more simple to measure occurrences of such crimes, for example in the news coverage, using this as a proxy for accountability.

If we look at this task more technically, we might describe it as inducing terms that describe the concept "crime" in a specific given domain. Again, this example serves illustrative purposes and is neither evaluated in a downstream task nor optimized in any form. Rather, it should serve as an example to illustrate how we apply the idea of shifting the induction in a desired direction using minimal effort.

We start with a rather sparse concept of crime, defined by only 11 terms related to crime in general (*Kriminalität (delinquency/crime), Verbrechen (crime), Verbrecher (criminal), Straftat (criminal act), Straftäter (perpetrator), Gefängnis (prison)*), but including a slight drift towards non-violent crimes (*Betrug (fraud), Korruption (corruption), Bestechung (bribery), Veruntreuung (embezzlement)*). Additionally, we also incorporate *Affäre (affair/scandal)*[11].

To demonstrate a slightly different approach than in the previous example (where we used an intermediate clustering step to find the next starting points) we use a different technique here. Since the the search part of the algorithm integrates random to increase the productivity, we may also leverage this fact by applying multiple scripted inductions and then aggregate the results. More precisely, we will produce 30 independent runs for the induction process and then merge the results before we apply a clustering based on the results[12].

The intuition behind this repeated procedure is that some of the runs will discover interesting sideways with the provided parameters during the search through the exerted influence of random. Since some of the runs are also erroneous, we seek to filter the results through aggregation and combination. The combination will be performed with the aforementioned clustering technique, with the modification that we preserve the influence of frequency for the clustering process. More precisely, if a word is found in several or even all of the 30 independent runs, it will have an according weight in the clustering process. On the other hand, this means that terms which are only found once

[11]Note that we also could inject the drifts through the definition of concrete starting points during the induction.

[12]This is not a very time-consuming procedure. The 30 passes for the scripted instructions were carried out in less than two minutes on a standard laptop

will have a diminished influence on the overall result of the clustering. The intention—based on experimental experience—is that the erroneously added terms will fall into the latter category.

But first, we will describe a single scripted run. Since we must direct the search to a point where we find crimes from the financial sector, we apply a steering via the initial starting point: *Finanzbranche (financial sector)*. We start for the first three recurrent runs[13] before we change the steering. We then first change the starting point to *Kreditbetrug (obtaining credit by false pretenses), Insidergeschäft (insider trading)* and afterwards to *Geldwäscherei (money laundering)*. Additionally, for the second and the third starting point, we decrease the influence of the shadow lexicon (containing uncertain information) by lowering its weight to 0.3 to add more caution or skepticism to the induction process. This is a reasonable step if we want to prevent false-positives and can afford this measurement when the induction is very productive.

Status before start:

- **Lexicon**: *Kriminalität, Verbrechen, Verbrecher, Straftat, Straftäter, Gefängnis, Affäre, Betrug, Korruption, Bestechung, Veruntreuung*
- **Shadow Lexicon**: *[]*
- **Exclusion Lexicon**:

In Table 6.7 we report the results for an accurate run[14]. The 119 newly added terms are nicely collected according to the intention to include crimes in the financial sector while avoiding terms for crimes related to physical or sexual violence. On the left-hand side, we see the starting points for the first run during the induction. Since reporting the full list of starting points would result in 120 data points[15], we refrain from displaying all of them but report all 20 iterations from the first run with *Finanzbranche (financial sector)* as initial starting point. Here we observe how the steering towards the finance sector is carried out implicitly in the second iteration (since none of the given 11 terms

[13] We also increased the number of iterations to 20. This parameter mainly influences "how far" the search will walk away from the initial starting point. Increasing this number of iteration makes sense if the re-sampled following starting points are meaningful combinations. This results in more candidates retrieved during the search phase before the assessment part of the algorithm is applied.

[14] Note that only the newly found terms are shown on the right-hand side, i.e., the original lexicon is not displayed here.

[15] We carry out 3 runs × 20 iterations for 3 different initial starting points: *Finanzbranche (financial sector)*; *Kreditbetrug (obtaining credit by false pretenses), Insidergeschäft (insider trading)*; *Geldwäscherei (money laundering)*

Re-Sampling during Search for first Run		
Run	It.	Starting Point Terms
1	1	*Finanzbranche*
1	2	Betrug, Finanzsektor
1	3	Veruntreuung, Betrügerei
1	4	Bestechung, Unterschlagung
1	5	Korruption, Bestechlichkeit
1	6	Betrug, Machtmissbrauch
1	7	Bestechung, Bestechlichkeit
1	8	Betrug, Amtsmissbrauch
1	9	Bestechung, ungetreu_Amtsführung
1	10	Veruntreuung, Amtsmissbrauch
1	11	Betrug, Urkundenfälschung
1	12	Bestechung, ungetreu_Geschäfts-besorgung
1	13	Betrug, ungetreu_Geschäfts-führung
1	14	Bestechung, ungetreu_Geschäfts-besorgung
1	15	Betrug, ungetreu_Geschäfts-führung
1	16	Veruntreuung, Urkundenfälschung
1	17	Betrug, ungetreu_Geschäfts-führung
1	18	Bestechung, Urkundenfälschung
1	19	Veruntreuung, ungetreu_Amtsführung
1	20	Betrug, ungetreu_Geschäfts-besorgung

Lexicon Content (only new terms) after Full Pass	
Abgabenbetrug	mehrfach_unwahr
aktiv_Beihilfe	mehrfach_Urkundenfälschung
aktiv_Bestechung	mehrfach_Veruntreuung
Amtsmissbrauch	Mehrwertsteuerbetrug
Anstiftung	Misswirtschaft
ausländisch_Amtsträger	mutmasslich_Beihilfe
Begünstigung	mutmasslich_Betrug
Beihilfe	mutmasslich_Steuerbetrug
Bestechlichkeit	mutmasslich_Veruntreuung
Bestechung_fremd	passiv_Bestechung
Bestechung_vorwerfen	Pfändungsbetrug
Betrug_mehrfach	Produktepiraterie
Betrug_ungetreu	Prozessbetrug
Betrug_Urkundenfälschung	qualifiziert_Geldwäscherei
Betrug_Veruntreuung	qualifiziert_ungetreu
Betrugsdelikt	qualifiziert_Veruntreuung
Betrugsversuch	schwer_Steuerdelikt
Betrügerei	Steuerbetrug
betrügerisch_Konkurs	Steuerdelikt
betrügerisch_Missbrauch	Steuerhinterziehung
Bilanzfälschung	Steuerhinterziehung_ermitteln
Bilanzmanipulation	Steuervergehen
Börsenmanipulation	Strafvereitelung
Datenverarbeitungsanlage	Terrorfinanzierung
Diebstahl_Betrug	Terrorismusfinanzierung
Dokumentenfälschung	ungetreu_Amtsführung
Erschleichung	ungetreu_Geschäftsbesorgung
falsch_Beurkundung	ungetreu_Geschäftsführung
Falschbeurkundung	Unterschlagung
Finanzdelikt	Untreue
Gehilfenschaft	Urkundendelikt
Geldwäsche	Urkundenfälschung
Geldwäscherei	Urkundenfälschung_schuldig
Geldwäscherei_anklagen	Urkundenfälschung_verurteilen
Geldwäscherei_ermitteln	Urkundenfälschung_vorwerfen
Geschäftsbesorgung	Vermögensminderung
gewerbsmässig_Betrug	versucht_Betrug
gewerbsmässig_Geldwäscherei	versucht_Erpressung
Gläubigerschädigung	Veruntreuung_Betrug
Gläubigerschädigung_durch	Veruntreuung_öffentlich
Hehlerei	Veruntreuungen
Hinterziehung_direkt	Vortat
illegal_Bereicherung	Vorteilsannahme
Insider-	Vorteilsgewährung
Insider-Geschäft	Vorteilsnahme
Insider-Handel	Warenfälschung
Insiderdelikt	wegen_Bestechung
Insidergeschäft	wegen_Betrug
Insiderhandel	wegen_Geldwäsche
Insidervergehen	wegen_Geldwäscherei
Konkursdelikt	wegen_gewerbsmässig
Kreditbetrug	wegen_Insiderhandel
kriminell_Organisation	wegen_Marktmanipulation
kriminell_Vereinigung	wegen_ungetreu
Kursmanipulation	wegen_Urkundenfälschung
Marktmanipulation	wegen_Verdacht
Marktmissbrauch	wegen_Veruntreuung
mehrfach_Betrug	Wertpapierbetrug
mehrfach_qualifiziert	wirtschaftlich_Nachrichtendienst
mehrfach_ungetreu	

TABLE 6.7: Exemplary re-sampling of terms for the starting point for the first 20 iterations during search and content of the lexicons after the full pass (No. 27). Starting point is defined by *Finanzbranche*.

for crime is in the vicinity of *Finanzbranche*, the first re-sampling is consisting of a close neighbor of *Finanzbranche* and a randomly drawn element of the lexicon, i.e., in this

case, *Betrug (fraud)*).

In Table 6.8 we report the result of a productive yet flawed induction process. Again, we present on the right-hand side only the 142 terms which have been added to the original lexicon. We observe that the induced lexicon contains many unwanted entries related to crimes of physical or sexual violence.

This is mirrored by the reported starting points on the left-hand side: in contrast to the pass reported in Table 6.7, the randomly chosen term from the original lexicon in the second iteration is *Gefängnis (prison)*. We trace the path of the search further, given the reported re-sampled starting points. The journey departs from *Gefängnis*, goes further from *hinter_Gitter (behind bars), Haft (custody), Verbrechen, Verbrecher* to *Mord (murder), Ermordung (homicide), vorsätzliche_Tötung (willful homicide), schwer_Körperverletzung (grievous bodily harm)* until it returns to rather business rooted crimes like *Betrug* or *ungetreu_Geschäftsführung (management impropriety)* and *ungetreu_Geschäftsbesorgung (corporate fraud)* where it starts to oscillate.

Naturally, and as we have shown in previous examples, one way to avoid the inclusion of such crimes is to put instances of them into the exclusion lexicon. But there are two drawbacks if we apply it in scenario like the ones described here. Firstly, we would have to check each of the 30 independent passes and its results. Secondly, the insertion of terms into the exclusion lexicon is a hard cut-off, in the sense that these terms are not taken into account during the full algorithm. This might also be prohibitive for terms that are related to the ones in the exclusion lexicon.

Therefore, we will apply now the technique depicted in the beginning of this section and aggregate the (uncorrected) resulting 30 lexicons from the 30 independent passes and cluster them. If a term appears in several of the induced lexicons, it will increase its weight during the clustering process. As seen before, we present the result of the clustering in the form of the nearest neighbors for the created centroids.

When we take Table 6.9 under closer investigation, we observe that none of the clusters represents the erroneous entries of crimes of physical or sexual violence which were integrated in some of the 30 passes we carried out for the largely automated induction. This demonstrates that the mitigation of their influence on the resulting lexical resource (the re-embedded and clustered lexicon in the form of the ten centroids) may be achieved by aggregation over several passes. In other words, the erroneous path that some of the induction passes have taken is corrected through aggregation of all passes before clustering. This means in turn, that such an approach is a viable alternative if less interactive usage is desired and possible labor-intensive post-editing should be avoided.

Re-Sampling during Search for first Run		
Run	It.	Starting Point Terms
1	1	*Finanzbranche*
1	2	Gefängnis, Finanzindustrie
1	3	Kriminalität, hinter_Gitter
1	4	Gefängnis, Haft
1	5	Verbrechen, hinter_Gitter
1	6	Verbrecher, Mord
1	7	Verbrechen, Ermordung
1	8	Veruntreuung, Mord
1	9	Betrug, vorsätzlich_Tötung
1	10	Veruntreuung, schwer_Körperverletzung
1	11	Betrug Körperverletzung
1	12	Veruntreuung, Urkundenfälschung
1	13	Bestechung, ungetreu_Geschäftsbesorgung
1	14	Betrug, ungetreu_Geschäftsführung
1	15	Veruntreuung, ungetreu_Geschäftsbesorgung
1	16	Betrug, Urkundenfälschung
1	17	Bestechung, ungetreu_Geschäftsbesorgung
1	18	Betrug, ungetreu_Geschäftsführung
1	19	Veruntreuung, Urkundenfälschung
1	20	Betrug, ungetreu_Geschäftsbesorgung

Lexicon Content after full pass	
Amtsmissbrauch	mehrfach_versucht
anderer_Delikt	mehrfach_Veruntreuung
Anklage_wegen	Mehrwertsteuerbetrug
Anstiftung	Misswirtschaft
ausländisch_Amtsträger	mutmasslich_Betrug
Begünstigung	mutmasslich_Steuerbetrug
Bestechlichkeit	mutmasslich_Veruntreuung
Bestechung_fremd	Nötigung
Betrug_anklagen	passiv_Bestechung
Betrug_mehrfach	Pfändungsbetrug
Betrug_ungetreu	Prozessbetrug
Betrug_Urkundenfälschung	qualifiziert_Freiheitsberaubung
Betrug_Veruntreuung	qualifiziert_Geldwäscherei
Betrug_vorwerfen	qualifiziert_ungetreu
betrügerisch_Konkurs	qualifiziert_Veruntreuung
betrügerisch_Machenschaft	räuberisch_Erpressung
betrügerisch_Missbrauch	Sachbeschädigung
Bilanzfälschung	Sachentziehung
Bilanzmanipulation	Schuldspruch_wegen
Börsenmanipulation	schwer_Körperverletzung
Datenverarbeitungsanlage	sexuell_Nötigung
Diebstahl_Betrug	Strafklage_wegen
Dokumentenfälschung	Straftatbestand
einfach_Körperverletzung	Strafvereitelung
Erpressung	Tatbestand
fahrlässig_Körperverletzung	ungetreu_Amtsführung
fahrlässig_Tötung	ungetreu_Geschäftsbesorgung
falsch_Anschuldigung	ungetreu_Geschäftsführung
falsch_Beurkundung	Unterschlagung
Falschbeurkundung	Untreue
Finanzdelikt	Urkundendelikt
Freiheitsberaubung	Urkundenfälschung
Gehilfenschaft	Urkundenfälschung_schuldig
Geldwäsche	Urkundenfälschung_verurteilen
Geldwäscherei	Urkundenfälschung_vorwerfen
Geldwäscherei_anklagen	Vergewaltigung_sexuell
Geldwäscherei_ermitteln	Vermögensminderung
Geschäftsbesorgung	Vermögensverminderung
gewerbsmässig_Betrug	Verschwendung_öffentlich
gewerbsmässig_Diebstahl	versucht_Betrug
gewerbsmässig_Geldwäscherei	versucht_Erpressung
Gläubigerbevorzugung	versucht_Nötigung
Gläubigerschädigung	versucht_schwer
Gläubigerschädigung_durch	versucht_Tötung
Hausfriedensbruch	versucht_vorsätzlich
Hehlerei	Veruntreuung_Betrug
Insider-Geschäft	Veruntreuung_öffentlich
Insider-Handel	Veruntreuungen
Insiderdelikt	Verurteilung_wegen
Insidergeschäft	vorsätzlich_Tötung
Insiderhandel	Vorteilsannahme
Insidervergehen	Vorteilsgewährung
Justizbehinderung	Vorteilsnahme
kaufmännisch_Gewerbe	Vorwurf_ungetreu
Kreditbetrug	Warenfälschung
Kursmanipulation	wegen_Bestechung
Körperverletzung	wegen_Betrug
Marktmanipulation	wegen_Freiheitsberaubung
mehrfach_Betrug	wegen_Gehilfenschaft
mehrfach_Diebstahl	wegen_Geldwäsche
mehrfach_Drohung	wegen_Geldwäscherei
mehrfach_Gefährdung	wegen_gewerbsmässig
mehrfach_Hausfriedensbruch	wegen_Körperverletzung
mehrfach_Körperverletzung	wegen_mehrfach
mehrfach_Nötigung	wegen_ungetreu
mehrfach_qualifiziert	wegen_Urkundenfälschung
mehrfach_Sachbeschädigung	wegen_Vergewaltigung
mehrfach_Tätlichkeit	wegen_Veruntreuung
mehrfach_ungetreu	Wertpapierbetrug
mehrfach_unwahr	wirtschaftlich_Nachrichtendienst
mehrfach_Urkundenfälschung	Zuhälterei

TABLE 6.8: Exemplary re-sampling of terms for the starting point for the first 20 iterations during search and content of the lexicons after the full pass (No. 7). Starting point is defined by *Finanzbranche*. Terms which are related to physical or sexual violence are underlined.

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	wegen_Betrug	0.8900
2	wegen_Veruntreuung	0.8633
3	wegen_Geldwäscherei	0.8590
4	wegen_Urkundenfälschung	0.8462
5	Urkundenfälschung	0.8298
6	Betrug_ungetreu	0.8252
7	wegen_ungetreu	0.8248
8	Veruntreuung	0.8207
9	qualifiziert_ungetreu	0.8195
10	Urkundenfälschung_verurteilen	0.7849

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Geldwäscherei_ermitteln	0.8136
2	Prozessbetrug	0.7944
3	wegen_Geldwäscherei	0.7930
4	Kreditbetrug	0.7886
5	wegen_Geldwäsche	0.7706
6	mutmasslich_Veruntreuung	0.7674
7	wegen_Betrug	0.7542
8	mutmasslich_Beihilfe	0.7538
9	Veruntreuung	0.7398
10	wegen_Verdacht	0.7352

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Urkundenfälschung	0.8991
2	ungetreu_Geschäftsbesorgung	0.8811
3	Veruntreuung	0.8638
4	gewerbsmässig_Betrug	0.8531
5	mehrfach_ungetreu	0.8520
6	ungetreu_Geschäftsführung	0.8519
7	Geschäftsbesorgung	0.8404
8	Gläubigerschädigung	0.8403
9	Falschbeurkundung	0.8329
10	Betrug_Veruntreuung	0.8283

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Insider-Handel	0.8121
2	Wertpapierbetrug	0.7925
3	Börsenmanipulation	0.7827
4	wegen_Insiderhandel	0.7825
5	wegen_Marktmanipulation	0.7485
6	Insiderhandel	0.7401
7	Wertschriftenbetrug	0.7377
8	wegen_Betrug	0.7015
9	Bilanzfälschung	0.6929
10	Prozessbetrug	0.6922

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Veruntreuung	0.8387
2	Urkundenfälschung	0.8305
3	Amtsmissbrauch	0.8163
4	Diebstahl_Betrug	0.8017
5	Betrug	0.7916
6	Unterschlagung	0.7916
7	ungetreu_Amtsführung	0.7890
8	qualifiziert_Veruntreuung	0.7816
9	Dokumentenfälschung	0.7783
10	Vorteilsannahme	0.7748

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Steuerhinterziehung	0.8912
2	Steuerbetrug	0.8822
3	Steuerdelikt	0.8285
4	Steuervergehen	0.7981
5	Hinterziehung	0.7742
6	aktiv_Beihilfe	0.7718
7	mutmasslich_Steuerbetrug	0.7706
8	Beihilfe	0.7692
9	Geldwäscherei	0.7449
10	Steuerhinterziehung_ermitteln	0.7299

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Bestechlichkeit	0.8581
2	Bestechung	0.8079
3	Begünstigung	0.7899
4	Vorteilsnahme	0.7724
5	Korruption	0.7536
6	Amtsmissbrauch	0.7474
7	Misswirtschaft	0.7466
8	Machtmissbrauch	0.7452
9	Veruntreuung	0.7307
10	Vetternwirtschaft	0.7225

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Insiderhandel	0.8905
2	Kursmanipulation	0.8445
3	Insidergeschäft	0.8308
4	Marktmanipulation	0.8232
5	Bilanzfälschung	0.7838
6	Insidervergehen	0.7639
7	Insider-Geschäft	0.7496
8	Geldwäscherei	0.7207
9	Bilanzmanipulation	0.7192
10	Insider-Handel	0.7060

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Urkundenfälschung	0.8682
2	mehrfach_Urkundenfälschung	0.8664
3	Betrug_mehrfach	0.8631
4	mehrfach_Betrug	0.8612
5	wegen_mehrfach	0.8602
6	mehrfach_Veruntreuung	0.8572
7	qualifiziert_Veruntreuung	0.8558
8	gewerbsmässig_Betrug	0.8525
9	Gehilfenschaft	0.8497
10	mehrfach_qualifiziert	0.8453

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Geldwäscherei	0.8975
2	Geldwäsche	0.8409
3	Warenfälschung	0.8072
4	Geldwäscherei_Korruption	0.7661
5	Steuerhinterziehung	0.7538
6	Terrorismusfinanzierung	0.7529
7	Terrorfinanzierung	0.7520
8	Geldwäscherei_anklagen	0.7477
9	Steuerdelikt	0.7345
10	Vortat	0.7344

TABLE 6.9: 10 most similar terms to the 10 centroids of the cluster model for the aggregation of the 30 lexicons for crime within the finance sector in the semantic space of the word2vec model, ordered by cosine similarity

A more detailed inspection allows us to see that we also obtain a clustering which infers sub-concepts of crimes in the financial sector. For example, centroid 4 and 8 point towards crimes which cover manipulation on stock markets where centroid 6 is essentially about crimes related to taxes. We also see further sub-concepts around bribery and corruption (centroid 7), or illegal business management practices (centroid 3), as well as more general clusters about embezzlement and fraud (centroid 1, 5, and 9).

Finally, money laundering is central for centroid 2 and 10, but the focus is subtly different: while centroid 2 is more about the financial aspects, money laundering in centroid 10 is even connected with financing of terrorism.

Of course, this kind of semantical clustering the lexicon provided into sub-concepts (or in this case of the aggregation of 30 lexicons from the 30 passes) does not come out of nowhere. The result of the clustering procedure reflects the connections (in the form of similarity) of meanings of terms which is modeled in the embedding, the model of distributional semantics which we learned from the news coverage in an unsupervised manner in the first place.

6.3 Inducing a Sentiment Lexicon

In this experiment we will derive a sentiment lexicon[16] from a small core seed and demonstrate how we disentangle semantic axes like sentiment which are interweaved in the embeddings.

One of the intentions to carry out this experiment is to demonstrate the applicability for a domain where lexicons for a comparison already exist and the concept that we aim to represent with the lexical resource is broadly studied (see Grimmer and Stewart (2013, p. 274), Neuendorf (2016, p. 146)). We will focus on a setting where we intend to expand a rather small lexicon—which could be created manually—with a fully automated work flow. More precisely, we will draw the seed from an pre-existing lexicon (Clematide and Klenner, 2010) and filter for the most frequent words (using statistics from `wordfreq`), for the negative and positive terms alike. We focus here on nouns and adjectives for each of which we draw 100 from each tonality to form the seed lexicon. The rationale behind

[16] Sentiment lexicons may be described as collections of terms that indicate negativity or positivity (see for example Wilson et al. (2005), Taboada et al. (2011)). Since we conceptualize the sentiment lexicon as two separate collections—one with terms of negative polarity and one with term of positive polarity—we could also say that we induced sentiment lexicon*s*. However, since it is common in this field to combine these two aspects of tonality in one lexical resource, we will use the term sentiment lexicon here. Of course, one needs to add the information of polarity for the terms in such a merged resource.

this is that we consider those words to be easily found in a manual or semi-automated way.

As our focus is on the automated expansion of small lexicons, we will evaluate the quality of this expansion through an ex-post annotation by three annotators. This means that we will evaluate the resulting list of terms which the program suggests as extension to the given seed lexicon. In addition to this, we will also report how many terms are new with respect to the source lexicon from Clematide and Klenner (2010) and how many terms are found within that lexicon. While the new terms may be seen as a valuable extension, we consider the induced terms that we find in an pre-existing curated resource as a proxy for the validity of the method, i.e., that it produces what it is intended to.

A second point we would like to demonstrate here is how we disentangle the intermixed information concerning sentiment in the embedding. As we have seen in Table 4.2 and Table 4.4, antonyms (with regard to their tonality) like *gut (good)* and *schlecht (bad)* are highly similar to each other in the embedding. This undermines simplistic approaches to expand sentiment lexicons where we just add close neighbors from present entries.

Therefore, we need to find a way to keep this information apart if we intend to induce a sentiment lexicon for which we obviously need to know if an induced term is of negative or positive polarity. In order to restrict the search for new terms and to influence the assessment step from the lexicon induction, we will make use of the mechanism to explicitly exclude other pre-existing lexical resources. Thus, we exclude the known terms of positive polarity in the search for new negative terms and vice versa.

Because we configure the search with a set of seed terms as a starting point, we use the same technique to cluster the lexicon as described in Chapter 5. More precisely, we cluster the current lexicon and derive the set of candidates from the nearest neighbors of the centroids we get from the clustering process[17]. With this list of sets of candidates at hand we initialize the lexicon induction. In Table 6.10 we list the starting points we assembled in this way for one run of lexicon induction concerning negative terms.

Since the final goal is to fully automate the induction and avoid as much noise as possible, we insert another layer of robustness. We have seen in Section 6.2.2 that we can benefit from the fact that most errors during induction are not systematic. Hence we counteract them through multiple execution with the same parameters and starting points. Because the lists of starting points for the individual runs of the lexicon induction vary in this experiment, we argue that terms which are found several times using different

[17]To increase variance we also re-combine these candidate lists, i.e., we first unite them all and randomly draw groups of three to form the starting points. Note that we keep nouns and adjectives apart in this process.

No.	Starting Point Terms
1	*unsicher, allein, krank*
2	*vermisst, verwirrt, schwierig*
3	*sinnlos, überflüssig, nervös*
4	*eingeschränkt, verlassen, merkwürdig*
5	*eigenartig, verschwunden, betreten*
6	*versteckt, problematisch, unnötig*
7	*betrunken, schlimm, wütend*
8	*furchtbar, seltsam, traurig*
9	*beschränkt, bedingt, schrecklich*
10	*Zorn, Schmerz, Verbrechen*
11	*Verwirrung, Haft, Überfall*
12	*Mangel, Verlust, Stress*
13	*Furcht, Bürgerkrieg, Verzweiflung*
14	*Unsicherheit, Betrug, Missbrauch*
15	*Schwäche, Krise, Elend*
16	*Krieg, Terror, Wut*
17	*Depression, Verkehrsunfall, Grauen*
18	*Wirtschaftskrise, Gefängnisstrafe, Chaos*
19	*Nachteil, Haftstrafe, Unfall*

TABLE 6.10: Starting points for the lexicon induction of negative sentiment terms, based on the clustering of the given seed lexicon

initializations are more likely to be correct. In other words, we consider the evidence higher if the terms show up in multiple runs[18].

Although we might strive to induce a much larger lexicon (leveraging chained induction processes in an iterative fashion), we evaluate the induction after the first run. We mainly restrict ourselves to this scenario due to the time-consuming manual evaluation (this aggregated induction already leads to a list of 827 (negative) and 739 (positive) new terms compared to the seed sets). Additionally, we will focus on the list of the induced negative terms.

These terms were evaluated by three annotators who had to rate them as being correct or wrong[19]. If the annotators disagreed in their votes, we take the majority vote for the term. Note that we excluded all terms for which at least one annotator chose the option to skip the annotation. This leaves us with 802 annotated terms. The averaged pairwise

[18]Similarly as in the experiment from Section 6.2.2, we argue, that we can indeed control the "amount of randomness" to make the automated induction process productive. But this random (one could also frame it as "creativity" in the induction process) may also be the cause for erroneous induction results—hence we have to counteract this if we apply the approach in a largely automated fashion. Otherwise we face the threat to propagate the error through the ongoing induction, leading possibly to much more errors.

[19]Additionally, the annotators were also allowed to skip the unit when they were unsure of the meaning of the word or if they had no clear-cut decision about its correctness.

F1-score for the annotation was 0.96 on the micro-level and 0.76 on the macro-level[20].

We give a quantitative overview of the results in Table 6.11. We see that the amount of correct entries is remarkably high with 96.4%. This means that we are able to reliably extend the seed by the fully automated induction process. However, we would still recommend to manually inspect the resulting extension of the lexicon since the errors should not propagated in downstream applications[21].

	Annotator A	Annotator B	Annotator C	Majority Vote (n=802)
correct	758 / 91.7 %	792 / 95.8 %	793 / 95.90%	773 / 96.4 %
wrong	62 / 7.5 %	18 / 2.2 %	32 / 3.90%	29 / 3.6 %
skip	7 / 0.85%	17 / 2.1 %	2 / 0.20%	

TABLE 6.11: Quantitative evaluation for the lexicon induction of negative sentiment terms in absolute counts and percent.

Out of the terms 773 correct terms we have induced, 399 are in the full lexicon of Clematide and Klenner (2010); this means in turn 374 terms are not yet included in the comprehensive lexicon, pointing to the potential of the approach to further extend already larger lexical resources.

Frequency Group	**Counts**	**Wrong**	**Correct**	**% Correct**
1	189	10	179	94.71%
2	122	9	113	92.62%
3	85	1	84	98.82%
4	59	2	57	96.61%
5	44	0	44	100.00%
6	59	1	58	98.31%
7	85	1	84	98.82%
8	29	1	28	96.55%
9	53	1	52	98.11%
10	77	3	74	96.10%
1-2	311	19	292	93.9%
3-10	491	10	481	98.0%

TABLE 6.12: Quantitative evaluation for the lexicon induction of negative sentiment terms per frequency group in absolute counts and percent.

[20] We report here the pairwise F1-score because it is in our view more appropriate for the scenario of a judgment concerning correctness. Since we have so many cases in which the annotators agreed on the term being correct, this results in a highly "skewed distribution" (many more "correct" votes than "wrong"). The κ statistics (Cohen, 1960) for inter-annotator agreement (note that the annotation in our case is a judgment on the system output and not an annotation on arbitrary words) are therefore misleadingly low with κ being 0.53. As an alternative, there exists also Gwet's AC1 (Gwet, 2014) which tries to overcome the paradoxon. Gwet's AC1 is in our case 0.96, similar to the micro-level F1 score, which might be more familiar to the reader. Therefore, we report pairwise F1 scores on micro and macro-level. See Feinstein and Cicchetti (1990) and Eugenio and Glass (2004) for a description of this paradoxon and Wongpakaran et al. (2013) for a comparison of Cohen's κ and Gwet's AC1.

[21] The manual annotation of the 827 terms took the annotators approximately 20 minutes.

In Figure 6.1 we report the results to test our assumption that we can in general trust the terms more which appear in the results of multiple runs. In order to investigate this claim, we evaluate the correctness of the terms with regard to their frequency over ten runs. To be clear, since the terms possibly appear in any of the results from the ten runs, the maximum is ten (frequency group 10) while a single occurrence would be evaluated in the frequency group 1. We see that the most errors stems mainly from the entries which occur only once or twice (or in other words: in the results of one or two runs). If we aim to avoid any manual control, we should cut off the aggregated results at this frequency. Nevertheless, it has to be mentioned that such a filtering is maybe too harsh. As we observe in Table 6.12, 92.6% of the terms which occurred twice are correct and even 94.7% of the terms which occurred once. This means that we would discard 311 candidates (189 + 122), 292 of them being correct if we applied this kind of filter, because of the 19 errors from these two large groups. But we confirm here that we mitigate the induction error if we apply the strategy of multiple runs and then aggregate the results.

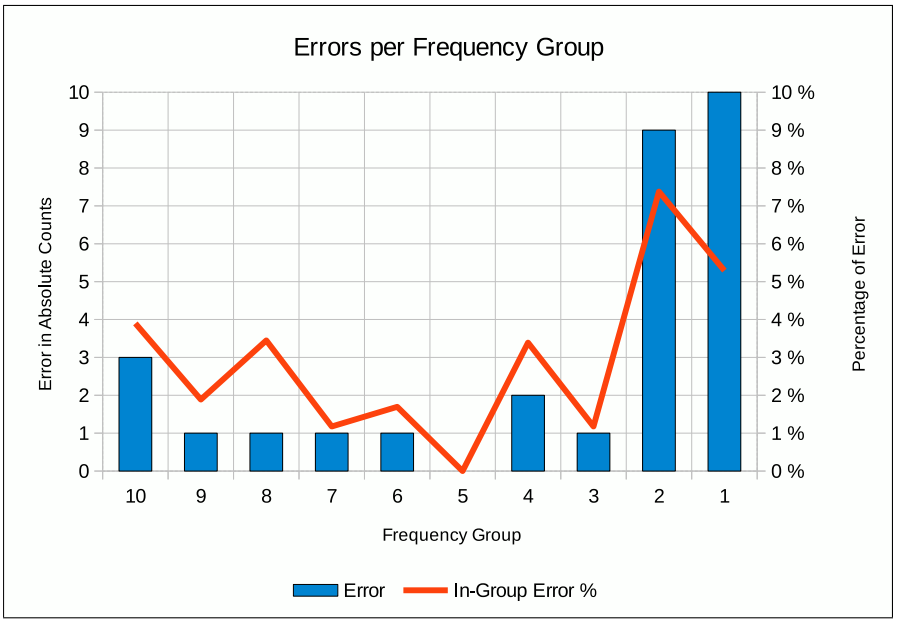


FIGURE 6.1: Distribution of errors in the aggregated result over frequency groups in absolute counts and as relative error rate

An interesting detail are the terms which occurred in all induction runs (frequency group 10) but which proved to be wrong: *Verwunderung*, *unvorstellbar*, and *Erleichterung*. If we for example look at *Verwunderung (astonishment)* and its representation in the model by inspecting its vicinity (given in Table 6.13), it becomes clearer why it breaks our premise. Since most terms in this list of nearest neighbors are themselves of negative tonality, this results in a systematic error. Since this error is not unsystematically

introduced by the random component, it is also not prohibited through the aggregation over multiple runs; the terms actually shows up in all the results.

15 most similar entries to "Verwunderung"		
Rank	Word	Similarity
1	Erstaunen	0.7782
2	Befremden	0.6861
3	Kopfschütteln	0.6549
4	Entsetzen	0.6398
5	Verblüffung	0.6248
6	Irritation	0.6202
7	Unverständnis	0.6148
8	Bedauern	0.6059
9	Unbehagen	0.6035
10	Stirnrunzeln	0.6015
11	Konsternation	0.5976
12	Empörung	0.5931
13	Bestürzung	0.5925
14	Skepsis	0.5914
15	Besorgnis	0.5808

TABLE 6.13: 15 most similar terms to *Verwunderung* in the semantic space of the word2vec model, ordered by cosine similarity

To sum up, we show that this kind of application of the approach is able to separate the entangled information we find in the embedding concerning the sentiment categories. Additionally, we also corroborate our finding that we mitigate the error of the induction if we perform multiple runs and aggregate the results. Although a hard cut-off would also decrease the resulting size of the extension substantially, there are multiple ways to extend the lexicon further, for example to start a new induction in the exact same way but with the defensively extended seed lexicon. If a manual correction of the automatically induced extension is carried out anyway, we could safely also include the terms which occur only once or twice since their accuracy is still between 92.6% and 94.7%, which is an acceptable error rate to keep the manual work on a low level.

6.4 Lexicons as Concepts for Classification and Detection

In this section, we refer to the resources that we have created for the document classification task discussed in Chapter 7 and for the framing detection task from Chapter 8.

One of the areas in which lexical-based classification systems are advantageous are settings which comprise of small data sets, many of which are also imbalanced with respect to the class distribution. Purely data-driven approaches suffer from insufficient data to generalize if the data set is small. This is especially true for approaches that model the units of analysis as bag of words (BoW). In such approaches that treat words as atomic symbols (i.e., words are not further analyzed and are conceptualized as the carrier of

information), only those words that occur in the training set can be modeled in the sense that the algorithm attributes a weight or probability for the different class labels.

The problem is often aggravated by a skewed data distribution causing the small classes to contain even lesser examples than they would under a uniform distribution of the data points over all classes. Lexical approaches may be beneficial for several reasons in this case:

One point is that the goal of a lexical resource is to generalize externally, i.e., capturing an idea through the aggregated information in the lexicon[22]. If we are able to extend the lexical resource and hence improve the prediction performance, we have fostered the generalization of the classifier, since it covers cases that are not represented in the training set. In other words, the lexical resource serves as a means to cover and describe a concept of interest (through the contained vocabulary). In contrast to data-driven BoW-approaches which learn such information from data, in lexical approaches the creation of the needed resource happens normally decoupled from the concrete task at hand. For example, manually curated collections of lexical triggers have been used for a long time (see Grimmer and Stewart (2013), Stone et al. (1966), Pennebaker et al. (2001)) for classification task.

An additional fact worth mentioning is that the generalization of a purely data-driven classifier is limited by its modeling. If the modeling is based on the bag-of-words approach, it is restricted to the calculation of features for words[23] that have been seen in the data. Of course, the weighting schema (e.g., TF-IDF) may help to guide and adjust the influence of the words. Nevertheless, the classic BoW-approaches are highly susceptible to badly represent classes for which only a small number of instances is available in the data set. This is no wonder, given the high probability that the variance in the small class is not covered properly due to the limited amount of examples.

One way to tackle this problem is the externalization of generalization, as we explain in the following. When we induce a lexicon for a class or label to be predicted (or the lexical manifestation of a concept), we create a point of reference against which we measure the units of analysis. More technically, we would perform a look-up on the tokens of the text and use the result for the prediction. The goal is hence to create lexicons which are then applied to retrieve a separable transformation of the unit of analysis at least for data points of different classes. In other words, if we perform look-ups in the lexicons, we should get distinct patterns of matches from the lexicons for the different classes. The simplest case is when we have a lexicon for each class we would like to separate. If

[22]Note that this is in principle independent from the data set and is a deductive process.

[23]In this context, we will refer to words but naturally also multi-word terms and bigrams or longer n-grams are subsumed under this term here.

we are able to induce such lexical resources which fulfill this requirement satisfactorily, we efficiently counteract the data skew for such tasks.

While the external generalization solves at least partly the problem of small data sets, it is also useful for tackling the problem of class imbalance. Class imbalance is a well studied problem and there are numerous approaches and methods which try to counteract the problem (cf. Haixiang et al. (2017) for a review). However, most of the approaches are not general enough or do not cover the case where we need to cope with both an imbalanced class distribution *and* a small data set. In contrast to this, if the external generalization is conducted in a manner that results in qualitatively and quantitatively equal coverage for the differently sized classes, the challenges of class imbalance are also addressed.

To support this claim, we will investigate three settings for documents classification and one for framing detection (see Chapter 7 and 8). To be clear, we emphasize here that we rely on the premise that the data points are separable considering their underlying concepts (i.e., the labels they were given, or the classes they have been assigned to). In other words, if we are able to measure accurately those properties of the text which actually cause the class membership, this enables us to perform classification on top of that measurement. The goal is therefore clear: to derive a lexical resource for each class to aptly measure the presence of the concept.

This last point leads also to a further advantage of lexical approaches. As we have already pointed out in Chapter 5, the problem for an over-represented residual class in data sets is not trivial. Especially when the only defining property of this class is the absence of certain phenomena. In other words, when the residual class represents the large portion of "everything else"—a class which may still need to be predicted during runtime, be it only to classify units of analysis as irrelevant—the appropriate steps to incorporate an apt model of this class into the classifier is far from trivial (see Chapter 8). However, this additional problem does not occur in cases where we already have an exhaustive set of labels which we need to take into account like the document classification task from Chapter 7. In this case, we just have to decide which of the specific labels from a given set is the correct or at least most probable one.

In the two following sections, we will furthermore describe the resources we have derived for a common document classification task and a framing detection task. The document classification is challenging because of the setting of skewed data distributions in relatively small data sets. The application for the framing detection task is built on a larger data set but faces the additional challenge of an over-represented residual class. The particular proposed solutions for the two tasks and their evaluation is given in the next two chapters. However, since those solutions are based on lexical resources specifically

created for this purpose, we describe those resources in the following sections. Moreover, the resources are induced using the same techniques as the examples given beforehand.

6.4.1 Lexicons for Document Classification

As document classification is one of the most common tasks, we will evaluate the proposed approach on such a given scenario.

Since we derive lexicons as lexical manifestations of the concepts underlying the classification labels in this case, we report here the contents of the lexicons and discuss briefly the steps that were taken to achieve the results. Additionally, we also give a partial overview[24] on the derived resources, considering one of the three classification tasks. We will therefore present and discuss also the re-embedded version of the lexicon to illustrate what is covered by the concept.

While we will concentrate here on the case for fine-grained classification of media documents concerning education for the in-detail parts, the described procedure was performed analogously for the two other domains (environment and traffic) as well.

6.4.1.1 Finding the Seed Core

We argued earlier that besides a given seed lexicon and the embedding, there are no further requirements for the lexicon induction. We follow the principle to allow for the inclusion of as much a priori information as possible—be it latently available information in annotated data, usage of further external resources, or through the injection of knowledge of the user. Hence, we we automatically derive this seed lexicon from the training set of the data provided for the classification task[25].

We therefore use the `SeedFinder` of the `ABCD` package to find words that are uncommonly often occurring. This is computed in relation to the other classes, to the corpus of all texts in the given data set, and also in relation to general a word frequency table, provided by `wordfreq` package (Speer et al., 2018). Additionally, we filter the words

[24]Since we develop 14 (3 + 5 + 6) lexicons for the classes modeled in the document classification, a discussion of all those resources would make the chapter longish and unpleasant to read. An encompassing description of the derived lexical resources as well as the illustrative form containing the nearest neighbors for the centroids of the concepts detectors is given in the Appendix B.

[25]An additional reason to follow this procedure is to increase the comparability to other, more data-driven approaches. More precisely, we make use of the given annotated data in the sense that we use it to automatically derive the seed lexicon—or at least make a usable rough draft of such a lexicon.

for nouns, adjectives, and verbs based on their PoS-tag[26] and subsequently filter them manually[27]

Seed Lexicon for Beruf/Berufsbildung	
Content after Manual Filtering	Filtered Words
Ausbildungsplatz, Automonteur, Berufsbildung, Berufswahlschule, Brückenangebot, Coiffeuse, Elektromonteur, Fachangestellte, Fähigkeitszeugnis, Floristin, Förster, Gärtner, interkantonale, Jobgarantie, Köchin, Kompetenzenbilanz, Kunststofftechnologe, KV-Lehre, KV-Lehrlinge, KV-Stelle, KV-Stellen, Lastwagenführer, Lehrabschluss, Lehrbetrieb, Lehrmeister, Lehrstellensuche, Maurer, Pflegeassistent, Polymechaniker, Praktikumsplatz, Sanitärmonteur, Schnupperlehre, Schnupperlehren, Schreiner, Sekundarklasse, Sportartikelverkäufer, Strassenbauer, Traumberuf, Volkswirtschaftsminister	Aargau, Bühler, Effretikon, Fasel, FHNW, Heterogenität, Ineichen, Junior, Kundert, Lange, Petermoos, Rennen, Solothurn, Zollinger

TABLE 6.14: Terms filtered out from the raw output of the SeedFinder and resulting seed lexicon to represent the subcategory *Beruf/Berufsbildung (Professions/Vocational training)*

In Table 6.14 we give an example of the filtering stage: if we investigate the filtered words, we find that most of them are named entities like names of persons (*Bühler, Fasel, Ineichen, Kundert, Lange, Zollinger*), names of locations *Aargau, Effretikon, Solothurn*, or institutions such as *FHNW (Fachhochschule Nordwestschweiz), Petermoos.* Additionally *Heterogenität* and *Junior* have also been filtered out. This example illustrates how such a filtering task could be on the one hand carried out manually without much effort, or, on the other hand, we could also further automate this by applying a named entity recognizer as an additional filter.

In Table 6.15 we report the number of words sieved by the SeedFinder, the number of words after manual filtering and the number of words that the actual lexicon for the

[26]Note that this is a rather rough filter and there is ample space for optimization or modification. The aim here is to show that even such a simple procedure produces an acceptable seed lexicon or at least a preliminary version of it through which we may sift manually. An interesting fact on its own is that this manual correction step normally triggers a lot of passive a priori knowledge of the user, i.e., if we have to correct a list of given terms for plausibility considering a certain concept, this step tends to activate our knowledge on that concept, leading to an easier access of memorized linked vocabulary.

[27]This manual filtering step consisted in many cases of the exclusion of named entities, especially family names and numeralia. These names were wrongly proposed by the `SeedFinder` mainly because of the high values for unexpectedness resulting from a statistical fall-back while using `wordfreq`, i.e., to take the class of lowest frequency for unknown terms. While there is good reason to exclude these names from the seed lexicon, these names are mostly stemming from prominently occurring actors in the media coverage of the specific subcategory.

Any kind of uninformed data-driven system would assign high weights to those names with high probability. Considering this aspect—and if the performance of the classification system is worth to sacrifice generalization—the removal of such data-specific idiosyncracies is rather limiting the classification performance. However, we carry out this step here since we are especially interested in creating a resource which covers a concept generally and not only in relation to the given data.

respective concept contains after the induction process (for the contents and its coverage see next section).

Subcategories	**Words in Seed**	**Filtered Seed**	**Words in Lexicon**
Beruf/Berufsbildung	53	39	257
Bildung/Schule/Hochschule	200	112	407
Wissenschaft/Forschung/Technik	202	124	305

TABLE 6.15: Number of words in the seed set (raw and manually filtered) and in the lexicon to represent the subcategories for the domain *Bildung (Education)*

It must be mentioned that the goal for this lexicon induction is not to create a resource that is as encompassing as possible given the model of distributional semantics. Rather, we use the annotated data we have for the task at hand to generate the seed lexicon and generalize this in the induction process. These steps are linked to our goal of external generalization but geared towards the concepts we find in the data. This means that we do not apply further techniques to inject more knowledge but rather just generalize through the expansion of the given (filtered) seed lexicon.

Subcategories	**Words in Seed**	**Filtered Seed**	**Words in Lexicon**
Abfall	18	16	178
Klima	201	177	291
Natur/Landschaft	155	78	230
Raumplanung	57	20	238
Tiere	204	133	425

TABLE 6.16: Number of words in the seed set (raw and manually filtered) and in the lexicon to represent the subcategories for the domain *Umwelt (Environment)*

In Table 6.16 we observe that after the expansion the difference in size between the smaller classes (*Abfall* and *Raumplanung* contain only a few instances; see Chapter 7) and the other classes is reduced through the lexicon induction, i.e., we create lexicons of similar size for those classes for which only a handful of documents is available.

Subcategories	**Words in Seed**	**Filtered Seed**	**Words in Lexicon**
Güterverkehr	12	28	123
Luftverkehr	170	95	441
Raumfahrt	104	61	220
Schienenverkehr/Bahn	151	82	275
Schifffahrt	55	17	182
Strassenverkehr	87	46	460

TABLE 6.17: Number of words in the seed set (raw and manually filtered) and in the lexicon to represent the subcategories for the domain *Verkehr (Traffic)*

In Table 6.17 we also report a case where the "filtered" version of the seed lexicon for *Güterverkehr* contains more words than the raw form. This is because we relaxed the parameters of the `SeedFinder` to create more candidates in a second run and merged this result with the raw first result them during the filter step.

After we have described the derived seed sets, we will now turn to the induced lexicons and especially also to the results of the re-embedding of the lexical resources, which enables us to control and inspect the lexical resource in the form it will be applied to the document classification task.

6.4.1.2 The Induced Lexicon and its Re-embedding: Concept Detectors

We have already described multiple methods (human intervention (with or without preliminary checks), random sampling, cluster based re-sampling) to steer and guide the lexical induction. And since we also want to refrain from recommending one as the best one—in fact, we would rather recommend mixing them all as it suits the case at hand—we will briefly report on the derived resources and then turn to the calculated concept detectors, i.e., the centroids of the re-embedded lexicon of the respective concept.

As mentioned before, we present here the lexicons for one of the classification tasks, i.e., for the classification of media texts from the domain education.

As an exemplary case we show partially the contents of the lexicon for *Beruf/Berufsbildung* after the induction process.The lexicon comprises of 257 terms from which we show the first 50 (in alphabetical order):

10_Schuljahr, Abitur, angehend, angehend_Lehrkraft, Anlehre, Ausbilderin, Ausbildner, Ausbildung, Ausbildung_absolvieren, Ausbildungsangebot, Ausbildungsgang, Ausbildungsplatz, Ausbildungsprogramm, Ausbildungsstätte, Ausbildungsweg, ausgebildet, Auszubildende, Autolackierer, Automatiker, Automech, Automechaniker, Automonteur, Bachelor-Abschluss, Bachelor-Studium, Bachelorabschluss, Bachelorstudium, Banklehre, Bauberuf, Bauspengler, Bauzeichner, Beruf, Beruf_ausüben, Beruf_erlernen, beruflich_Ausbildung, beruflich_Grundbildung, beruflich_Weiterbildung, Berufs-, Berufsabschluss, Berufsalltag, Berufsausbildung, berufsbegleitend, Berufsbegleitend, berufsbegleitend_Ausbildung, berufsbegleitend_Weiterbildung, Berufsbild, Berufsbildung,...

Before investigating its content more closely, we would like to point out that the listing of this lexicon has only illustrative purposes. We attempt to give a more concise view on this resource with the vicinity of the centroids of a cluster model of it. (see Table 6.18)

The lexicon for the class Beruf/Berufsbildung contains, trivially, many professions, many mentions of different version of professional training (*Weiterbildung, Weiterbildungsangebot, Weiterbildungskurs, Weiterbildungsmöglichkeit, zweit_Bildungsweg, Zweitausbildung*, just to name a few) but interestingly also many mentions of terms related to studying at a university institution (for example *ETH-Studium, Master-Abschluss, Fachhochschulstudium, Fernstudium, Zweitstudium*). The question arises if these terms should be present in the lexicon or not.

It should be kept in mind that we create the resource to solve the task of classification of articles which we frame as the detection of the concepts in articles. Therefore, our detection approach must create signals that are strong enough to separate the different concepts or classes, i.e., *Beruf/Berufsbildung (Professions/Vocational training)* vs. *Bildung/Schule/Hochschule (Education/School/University)* vs. *Wissenschaft/-Forschung/Technologie (Science/Research/Technology).*

These concepts exhibit several points where there is an overlap. For example, a *University*[28] is an institution to study and learn; it is also part of the educational system; and it is also one of the main institutions renowned for science, research, and technology. There are different ways to answer to the question how we deal with this ambiguity concerning the lexical resources.

On the one hand, we could decide not to integrate the term *University*[29] in any lexicon, because of the false positives that it will create automatically in a look-up scenario (remember that we are facing a single-label task). On the other hand, we could delete the term from only one or several lexicons for which we would estimate the contribution to be less central. A third alternative is to keep it in all the lexicons in which it is occurring, i.e., not imposing any disjunctivity requirement on the lexicons.

There are several reasons to follow the third path. Firstly, since *University* in one or another form is actually a lexical signal for all of the classes to predict, we should not exclude it from any of the lexicons. Secondly, we think the decision how to handle the membership in multiple lexicons should be delegated to the application using the lexical resources. Whether the ambiguity is solved with hard rules, weights (i.e., statistics and/or probability) or even context-specific means is rather a step to be considered at application building time. Thirdly, while integrating this indecisiveness for the lexical resources may seem to complicate the application step, this turns out to be useful for mutually non-exclusive multi-labeling scenarios[30]. Lastly, if we apply any transformations on the raw lexical resources to make them (more) suitable for an application, the inclusion of these terms may even be beneficial since they help to sharpen the context of the transformation. Therefore we will find terms related to *University* in all three lexicons we induced to carry out the classification task.

Since we think that the presentation of all induced lexicons as lists of words is not the best format to gain insights, we will offer another view on the generated resources

[28]Note that we do not refer only to the word *University* but rather on its meaning.

[29]Consequentially, we also refer here to group of terms related to this point of overlap; naturally, the term *University* itself is a member of this group.

[30]Consider such cases where we would have to assign not only one label but a set of labels to a document. If a term like *University* is a good trigger for a signaling of multiple classes, we should refrain from assigning such a term to only one lexicon.

that we have used before in this work. To get a more comprehensive presentation of the concept covered by the lexicon, we perform a clustering step and then study the resulting centroids (see also Chapter 5). This presentation form has also several minor advantages considering the intended purpose of the resource:

- The view is closer to the application: since we transform the resources anyway in this manner for the downstream application of document classification, the result of the transformation shows more closely what we actually integrate into the application.
- The view offers insights into the structure of the concept: When we analyze the direct vicinity of a centroid, we often detect the sub-concept that formed the cluster of that centroid. This means that the results from the clustering process are an inferred structure from the given lexicon. We refer to them as sub-concepts.
- The view allows for more detailed analyses: the inferred structure of the sub-concepts may additionally help us to either fine-tune the classifier or use the fine-grained information of the clustered lexicon to deliver even more detailed analyses.

We will now turn to a closer examination of the re-embedded and clustered lexicons which we have derived for the document classification task. Especially for the document classification for articles about education, we will provide a description of the inferred structure in the form of clusters and their respective centroids. The full listing of this kind of view on the centroids for all the other lexicons is given in Appendix B.

In Table 6.18 we see the results of the clustering process on the lexicon for the class Beruf/Berufsbildung. While we find terms for education in general near the centroid 2, we see the connections of the class to the *University* aspect that we mentioned above in centroid 1.

Centroid 3 represents several different paths in the Swiss education system (*Matur, Matura* vs. *Berufsmatura*, vs. *kaufmännisch_Lehre, kaufmännisch_Ausbildung*, etc.) Centroid 4 and 5 are rather focused on the non-academic path of the dual Swiss system (i.e., the vocational education and training or "apprenticeship"). Centroid 5 also comprises of several terms related to the point in time when the compulsory school education is over (i.e., *Brückenangebot, 10_Schuljahr, Schulabgänger, Schulabgängerin*).

In the centroids 6 and 9 we find the professions clustered. And indeed, we find the gender specific clichés we would expect given the results from research on bias in embeddings (eg., Bolukbasi et al. (2016)).

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Studium	0.8711
2	Lizenziat	0.8150
3	Doktorat	0.8102
4	Wirtschaftsstudium	0.7970
5	Lizentiat	0.7692
6	Geschichtsstudium	0.7654
7	Zusatzstudium	0.7609
8	Jusstudium	0.7606
9	Matura	0.7599
10	Zweitstudium	0.7582

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Ausbildung	0.8670
2	Grundausbildung	0.8214
3	Lehrgang	0.7995
4	Zusatzausbildung	0.7838
5	ausbilden	0.7410
6	einjährig_Ausbildung	0.7401
7	zweijährig_Ausbildung	0.7398
8	ausgebildet	0.7389
9	weiterbilden	0.7387
10	berufsbegleitend	0.7381

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Matura	0.8211
2	kaufmännisch_Lehre	0.8193
3	Handelsschule	0.8045
4	KV-Lehre	0.8016
5	Handelsdiplom	0.7850
6	kaufmännisch_Ausbildung	0.7782
7	Praktikum	0.7762
8	Matur	0.7650
9	Berufsmatura	0.7455
10	Schreinerlehre	0.7452

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	vierjährig_Lehre	0.7720
2	dreijährig_Lehre	0.7696
3	Lehrling	0.7461
4	Polygrafen	0.7315
5	Verkaufslehre	0.7286
6	Schnupperlehre	0.7246
7	Praktikum	0.7245
8	Ausbildung	0.7202
9	Lehrabschlussprüfung	0.7199
10	viert_Lehrjahr	0.7137

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Lehrstelle	0.8553
2	Ausbildungsplatz	0.8083
3	Brückenangebot	0.8018
4	Schulabgänger	0.7990
5	Lehrbetrieb	0.7899
6	Lernende	0.7803
7	10_Schuljahr	0.7556
8	Praktikumsplatz	0.7481
9	Schulabgängerin	0.7462
10	Auszubildende	0.7446

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Automechaniker	0.8096
2	Elektromonteur	0.8088
3	Schreiner	0.7887
4	Polymechaniker	0.7741
5	Hochbauzeichner	0.7488
6	Automatiker	0.7323
7	Betriebspraktiker	0.7296
8	Bauzeichner	0.7277
9	Sanitärmonteur	0.7216
10	Landschaftsgärtner	0.7206

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Berufslehre	0.8280
2	Berufsmaturität	0.8063
3	Berufsausbildung	0.8041
4	Berufsmatura	0.8020
5	Berufsmatur	0.8017
6	Fachhochschulstudium	0.7891
7	berufsbegleitend	0.7835
8	Matur	0.7804
9	Ausbildung	0.7789
10	Matura	0.7760

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Weiterbildung	0.8249
2	Weiterbildungsangebot	0.8080
3	Ausbildungsgang	0.7920
4	Lehrgang	0.7670
5	hoch_Fachschule	0.7439
6	Weiterbildungsmöglichkeit	0.7365
7	Ausbildungsangebot	0.7353
8	berufsbegleitend	0.7332
9	Studiengang	0.7315
10	Bildungsgang	0.7296

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	kaufmännisch_Angestellte	0.8076
2	Kauffrau	0.7997
3	diplomierte	0.7885
4	Coiffeuse	0.7645
5	Fachfrau_Betreuung	0.7475
6	Floristin	0.7361
7	gelernt	0.7240
8	Autolackiererin	0.7180
9	Hochbauzeichnerin	0.7176
10	Primarlehrerin	0.7069

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Beruf	0.7895
2	Berufsfeld	0.7621
3	Lehrerberuf	0.7606
4	Lehrberuf	0.7602
5	handwerklich_Beruf	0.7170
6	Berufsleute	0.7158
7	Berufswelt	0.7104
8	Berufslehre	0.7102
9	kaufmännisch_Bereich	0.7072
10	Berufsleben	0.7014

TABLE 6.18: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for *Beruf/Berufsbildung (Professions/Vocational training)* in the semantic space of the word2vec model, ordered by cosine similarity

In centroid 7 we see several ways of further vocational training as well as possibilities to higher education, for example at a school of applied sciences (*Fachhochschule*) after having earned the qualification for university entrance (but limited to a specific branch which is related to the profession one is working in; i.e. getting the *Berufsmatura*).

Centroid 8 is about further education and advanced training in general while the focus from centroid 10 is more on professions (as a generic concepts) and related terms.

In Table 6.19 and 6.20, we find the results of the clustering on the lexical resources we have induced for the two remaining classes (*Bildung, Schule, Hochschule* and *Wissenschaft, Forschung, Technologie*).

For *Bildung, Schule, Hochschule*, we find in Table 6.19 centroids representing specific parts of the educational system (as intended).

In centroid 1 we find the different degree programs for students on a university or other schools of higher education. The political instances that are involved in steering, implememtation and development of the educational system are found in centroid 2.

In centroid 3, 8, and 10, the different institutions for higher education are mirrored: while we find the *Universität (university)* and the *ETH (Federal Institute of Technology)* in cluster 10, we see the schools of applied sciences represented by centroid 8. Centroid 3 represents especially the school of higher education for teachers of the several stages in the educational system (i.e., *pädagogisch_Hochschule*, *PHZH*, *PH_Zürich*). Centroid 4 represents the stages of the compulsory school education, i.e., the structure.

We have already mentioned the references to the university as an institution in centroid 10. In contrast to this, centroid 5 also represents partially the concept of a University but captures it more via the persons and specifically their functions. So we find Student, Dozent(lecturer), Professor, or Doktorand (doctoral student) as well.

Centroid 6 is focused the *Gymnasium (secondary school/high school)* that allows for entrance at the University level.

In centroid 7, younger pupils and even *Kindergarten* are represented. It also contains structural terms (*Primarschule, Sekundarschule, Mittelstufe, Oberstufe*).

Interestingly, in centroid 9, we find terms which are highly similar to the ones from centroid 7 in Table 6.18, which illustrates nicely the overlap of the two concepts for the classes which orbits around the sub-concept of *Berufsmaturität*. This is no surprise, given that it is one of the transition points in the Swiss dual system of education where people should be enabled to make the decision for higher education after having learned a profession via apprenticeship/vocational training.

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Studiengang	0.8779
2	Studierende	0.8237
3	Fachhochschule	0.8209
4	Bachelor-	0.8144
5	Bachelor	0.8125
6	Masterstudiengang	0.7992
7	Masterstudium	0.7926
8	Studienrichtung	0.7894
9	Absolvent	0.7867
10	Masterstufe	0.7813

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Volksschule	0.7286
2	Bildungsdirektion	0.7120
3	Erziehungsdirektorenkonferenz	0.7012
4	Mittelschule	0.6788
5	Lehrkraft	0.6775
6	Bildungsrat	0.6684
7	Lehrperson	0.6516
8	Lehrplan	0.6510
9	Mindestpensen	0.6501
10	Lehrpersonenkonferenz	0.6476

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	pädagogisch_Hochschule	0.8392
2	Fachhochschule	0.8011
3	Studiengang	0.7796
4	Angewandte_Linguistik	0.7548
5	Berufsschule	0.7480
6	hoch_Fachschule	0.7461
7	Mittelschule	0.7461
8	Lehrgang	0.7454
9	PHZH	0.7432
10	PH_Zürich	0.7395

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Oberstufe	0.8695
2	Primarschule	0.8528
3	Sekundarschule	0.8339
4	Sekundarstufe	0.8273
5	Volksschule	0.8210
6	Lehrperson	0.8205
7	Lehrkraft	0.8133
8	Unterstufe	0.8005
9	Mittelstufe	0.7991
10	Grundstufe	0.7977

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Universität	0.8003
2	Student	0.7982
3	Uni	0.7872
4	Dozent	0.7650
5	Professor	0.7441
6	Doktorand	0.7437
7	Studierende	0.7212
8	Harvard	0.7167
9	Studentin	0.7129
10	Eliteuniversität	0.6997

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Gymnasium	0.8701
2	Mittelschule	0.8493
3	Langgymnasium	0.8275
4	Langzeitgymnasium	0.8039
5	Kurzzeitgymnasium	0.8005
6	Sekundarschule	0.8003
7	Aufnahmeprüfung	0.7918
8	Kurzgymnasium	0.7898
9	Sekundarstufe	0.7661
10	Gymi	0.7607

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Schüler	0.8346
2	Primarschüler	0.8345
3	Kindergarten	0.8235
4	Unterricht	0.8227
5	Oberstufe	0.8213
6	Primarschule	0.8176
7	Mittelstufe	0.8118
8	Sekundarschule	0.7924
9	Schule	0.7920
10	Klassenzimmer	0.7884

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Zürcher_Hochschule	0.8525
2	Hochschule	0.8378
3	angewandte_Wissenschaft	0.8019
4	Wädenswil_ZHAW	0.7986
5	Kunst_ZHdK	0.7961
6	Kunst_HGKZ	0.7926
7	ZHAW	0.7922
8	Kunst_ZHDK	0.7822
9	ZHdK	0.7723
10	Angewandte_Wissenschaft	0.7634

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Berufsmaturität	0.7952
2	Fachmittelschule	0.7943
3	Gymnasium	0.7875
4	Mittelschule	0.7870
5	Berufslehre	0.7717
6	Matur	0.7716
7	Fachmaturität	0.7711
8	Maturität	0.7598
9	Berufsmatur	0.7558
10	Berufsmittelschule	0.7544

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Uni	0.8452
2	Fakultät	0.8173
3	ETH	0.8005
4	medizinisch_Fakultät	0.7896
5	Uni_Zürich	0.7855
6	Universität	0.7698
7	Universität_Zürich	0.7678
8	Fachhochschule	0.7678
9	Studierende	0.7580
10	Doktorand	0.7549

TABLE 6.19: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for *Bildung/Schule/Hochschule (Education/School/University)* in the semantic space of the word2vec model, ordered by cosine similarity

Table 6.20 show that the embedded clustered lexicon is more clearly separated from the other two classes (despite of the other two containing mentions of the university as shown before).

We find some clusters which represent research in general (centroid 7 and 8) and also a cluster which is more specific, geared towards the results of research (centroid 1).

Additionally, we find here more structuring for several fields of science or even disciplines. While centroid 2 represents different aspects of neuroscience, centroid 3 is more closely centered around biology (with some tendency to physics). In centroid 5 we find the intersection of biology and technological approaches, and in centroid 9 mathematics is central, as is (quantum) physics in centroid 10. Interestingly, other fields of science besides natural science seem to be grouped in cluster 4.

We also observe that the technological aspect—which is also very important for this category—is represented densely in centroid 6.

To sum up, we were able to easily infer topical sub-concepts in all of the re-embedded and clustered lexical resources[31]. As mentioned beforehand, the data-driven identification of the sub-concepts is an interesting side effect of the clustering which allows us to point back to the specific centroids (and hence sub-concepts) if desired. Trivially, it also allows for an inspection which of the (automatically inferred) sub-concepts of the lexicon are especially present (or absent) in the documents we want to analyze. For the basic task of document classification we will however generalize over the sub-concepts in the sense that we sum up their contribution to the overarching class label as described in Chapter 5.

[31] One should be aware that we might also detect that important parts of the concept are lacking. This is either the case when we do not provide enough freedom to the clustering process in the form of number of clusters, or when we do not have enough terms in our lexicon which point to this sub-concepts. While we may just increase the number of clusters for the former case, in the latter case, we might consider to enhance the lexical resource with further induction steps geared towards that goal.

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Forschungsergebnis	0.8064
2	wissenschaftlich	0.7992
3	wissenschaftlich_Erkenntnis	0.7466
4	empirisch	0.7421
5	Forschungsarbeit	0.7247
6	Forschungsresultat	0.7092
7	Forschungserkenntnis	0.7048
8	neurowissenschaftlich	0.6964
9	Forschungsansatz	0.6902
10	Hirnforschung	0.6895

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Hirnforschung	0.7917
2	Biologie	0.7884
3	Neurowissenschaft	0.7649
4	Anthropologie	0.7609
5	Wissenschaft	0.7575
6	Neurobiologie	0.7471
7	Naturwissenschaft	0.7262
8	naturwissenschaftlich	0.7241
9	Psychologie	0.7147
10	Geisteswissenschaft	0.7136

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Zellbiologie	0.8018
2	Biochemie	0.7901
3	Mikrobiologie	0.7861
4	Verhaltensbiologie	0.7769
5	Pflanzenbiologie	0.7727
6	Molekularbiologie	0.7686
7	Universität_Bern	0.7638
8	theoretisch_Physik	0.7566
9	Umweltphysik	0.7515
10	Pflanzenwissenschaft	0.7478

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Soziologie	0.8664
2	Rechtswissenschaft	0.7979
3	Germanistik	0.7972
4	Wirtschaftswissenschaft	0.7935
5	Psychologie	0.7914
6	Literaturwissenschaft	0.7896
7	Ethnologie	0.7833
8	Kulturwissenschaft	0.7819
9	Islamwissenschaft	0.7786
10	Religionswissenschaft	0.7782

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Biotechnologie	0.8015
2	Materialwissenschaft	0.7978
3	Life_Sciences	0.7393
4	Life-Sciences	0.7333
5	Biowissenschaften	0.7333
6	Life-Science	0.7322
7	Verfahrenstechnik	0.7311
8	Pharmazie	0.7291
9	Informatik	0.7198
10	Elektrotechnik	0.7176

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Technologie	0.8249
2	technologisch	0.7523
3	Innovation	0.7359
4	technisch_Innovation	0.7281
5	Informationstechnologie	0.7135
6	technologisch_Fortschritt	0.7077
7	technologisch_Entwicklung	0.7043
8	modern_Technologie	0.6893
9	technisch_Fortschritt	0.6781
10	Technik	0.6781

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Grundlagenforschung	0.7833
2	Forschung	0.7435
3	Forschungsgebiet	0.7315
4	regenerativ_Medizin	0.7235
5	Forschungsbereich	0.7171
6	Forschungsschwerpunkt	0.7054
7	forschen	0.7050
8	Systembiologie	0.7022
9	Biosysteme	0.7017
10	Neurowissenschaft	0.7010

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Grundlagenforschung	0.8369
2	Forschung	0.8165
3	angewandte_Forschung	0.7449
4	Nationalfond	0.7271
5	anwendungsorientiert_Forschung	0.7182
6	anwendungsorientiert	0.7181
7	Spitzenforschung	0.7012
8	Forschungsprojekt	0.6996
9	Forschungsbereich	0.6970
10	universitär	0.6961

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Mathematik	0.8522
2	Naturwissenschaft	0.8220
3	Chemie_Physik	0.7768
4	Mathematik_Naturwissenschaft	0.7562
5	Biologie_Chemie	0.7555
6	Mathematik_Deutsch	0.7550
7	Physik	0.7484
8	Biologie	0.7403
9	Fach_Deutsch	0.7403
10	Physik_Chemie	0.7374

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	physikalisch	0.7593
2	Quantentheorie	0.7445
3	Quantenmechanik	0.7444
4	menschlich_Gehirn	0.7392
5	molekular	0.7237
6	Hochtemperatursupraleitung	0.7221
7	Quantenphysik	0.7161
8	biologisch_Evolution	0.7075
9	Supraleitung	0.7046
10	Elektromagnetismus	0.7044

TABLE 6.20: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for *Wissenschaft/Forschung/Technologie (Science/Research/Technology)* in the semantic space of the word2vec model, ordered by cosine similarity

6.4.2 Lexicons for Framing Detection

While document classification is a more common task, we report in this section on the results for the lexical induction process as a primary stage for the framing analysis task addressed in Chapter 8.

In order to focus on the lexicon induction process in this chapter, we will leave out the steps[32] that derive the core lexicons for the different frames, and report in more detail on this in Chapter 8.

To be clear, we report in this chapter quantitatively on the outcome of the lexicon induction process, and show via the re-embedding of the lexicon what is actually represented in the resources. We apply these resources later in Chapter 8, where we will also take a closer look at the outcome of the framing analysis itself. This means that we do not report measurements on the extrinsic task in this chapter, but we aim to prepare the stage so that the description of the experiments and the evaluation in Chapter 8 are concise and sleek.

But firstly, we will give a short introduction to the task at hand and provide a minimalist explanation by a set of examples to introduce the topic. This is at least required to understand the lexically induced and re-embedded concepts that we present in this section.

6.4.2.1 Legitimacy Frames

For the framing detection task, we would like to detect the presence (or claimed absence) of accountability of new forms of governance (see also Amsler et al. (2016) and Wüest et al. (2017)). Therefore, we analyze text documents from mass media outlets about the entities of interest and furthermore aim at detecting frames. We conceptualize frames as schemata of interpretation (cf. Goffman (1974, p. 21)) that refer to the possible sources of democratic legitimacy. In other words, we are interested in finding out how the decision-making process of those forms of governance is described. For instance, there might be an emphasis on the aspect of the participation (i.e., if citizens were involved). Another example would be an article that contains the critique of a lack of effectiveness considering measurements for environment protection which refer to international treaties.

[32] As it will be shown in Chapter 8, the labels to predict and the unit of analysis differ in this case. But the basic principles are analogous to the ones described in the previous section about finding the seed core. More precisely, we used what we had as supervision, meaning, we extracted the seed lexicon from passages of text, that were identified as the "core of the frame"

The sources for democratic legitimacy may be input-oriented, output-oriented, or throughput-oriented (see Schmidt (2013)). In simple words, they are differentiated as follows:

- **Input-oriented legitimacy frames** point to aspects relating to who was involved in the decision-making process. There is a distinction on a more fine-grained level which distinguishes aspects of representation (elected representatives), participation (involvement of citizens), deliberation (careful consideration and/or discussion), epistemicity (involvement of experts/scientists), and stakeholder inclusion.
- **Throughput-oriented legitimacy frames** are present when the text focuses on aspects that describe attributes of the process itself. On the fine-grained level, we distinguish transparency (publicly available and disseminated information on the process), accountability (processes are controllable and correctable) and legality.
- **Output-oriented legitimacy frames** represent aspect of the results of the political decision-making process. We differentiate here between efficiency frames (measures are efficient with regard to costs or time) and efficacy frames (measures lead to changes and solve the problems at stake).

For this research project, the entities of interest are diverse: we examine transgovernmental networks, international treaties, regulatory bodies, or public transport organizations of metropolitan regions by the media coverage which we retrieve about them. For example, an article which claims that the process of re-organizing the public transport system is too slow. This is an instance of an efficiency frame, or, more precisely, the lack of it.

An example of an international treaty is the Kyoto Protocol. In the respective case we would consider utterances such as *Without the ratification of the protocol by the United States, the intended outcome is in danger.* as an efficacy frame. Again, also here, the criticism is on the lack of efficacy.

6.4.2.2 Lexicon Induction and Re-embedding

We present in this section a purely quantitative overview of the lexicons that we induced for each of the given ten framing categories. Furthermore we give a more detailed description for three specific frames (representation, efficacy, throughput) in the forms of the re-embedded lexical resources, visualized through the centroids of the cluster process.

In Table 6.21 we see that we created a lexicon for each category consisting between 212 and 414 entries. While the number of words is not directly related to the quality

of the detectors we derive from the lexicon by means of the clustering process, it is nevertheless important to make sure that the coverage of the lexicon includes what has been conceptualized.

Coarse-Grained Category	**Fine-Grained Category**	**Words in Lexicon**
Input	Deliberation	398
	Epistemic	329
	Participation	370
	Representation	331
	Stakeholder	353
Throughput	Accountability	276
	Legality	414
	Transparency	329
Output	Efficacy	383
	Efficiency	212

TABLE 6.21: Number of words in the lexicon for the frame categories

As before, we allow for such an inspection of the detectors through the assessment of their vicinities. We present one example for each coarse-grained category of frames of democratic legitimacy: Table 6.22 shows the centroids for the lexicon for *Representation* (as an instance of input legitimacy), Table 6.23 shows the centroids for the lexicon for *Efficacy* (as an instance of output legitimacy), and Table 6.24 shows the centroids for the lexicon for *Transparency* (as an instance of throughput legitimacy).

When inspecting the detectors for *Representation* frames in Table 6.22, we are able to clearly identify different groups of representatives or institutions that are represented[33]. Centroid 1 represents the institutions of the United States as well as the main parties. Centroid 2 represents Swiss parties, while centroid 6 represents German parties and coalitions. Parties are also addressed more generically in centroid 4. Institutional functions and positions such as president and prime minister or other ministerial posts are represented in centroid 3, 5, and 10. As Switzerland is institutionally different, and the names of governance bodies also differ, we see them reflected in centroid 7. In centroid 8 we find a grouping of terms referring to the parliaments and the parliamentarians in the different countries and political systems. Finally, in centroid 9, there are references to the political entities of the Kanton Zurich and Zurich as a city (as a municipality, the lowest level of the Swiss federalist system).

[33]A wide range of entities was studied for the framing analysis, but we tried nevertheless to derive a lexicon for the frames which was applied across all types of entities. Additionally, we attempted to cover the legitimacy frames for different perspectives, i.e., different countries with different political systems (please refer also to Chapter 8 for more details). This in turn results in frame lexicons which do not focus on national peculiarities of the political systems. However, they are partly represented in specific centroids.

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Repräsentantenhaus	0.8173
2	Republikaner	0.8109
3	Demokrat	0.7973
4	Senat	0.7881
5	Senator	0.7727
6	Harry_Reid	0.7277
7	Kongress	0.7188
8	republikanisch	0.7104
9	weiß_Haus	0.7039
10	US-Senat	0.6944

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	SVP	0.8443
2	CVP	0.8412
3	Grünliberale	0.8342
4	GLP	0.8320
5	SP	0.8319
6	BDP	0.8093
7	Freisinnige	0.7983
8	EVP	0.7746
9	Bürgerliche	0.7717
10	EDU	0.7672

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Minister	0.8560
2	Wirtschaftsminister	0.7697
3	Innenminister	0.7511
4	Umweltminister	0.7310
5	Staatssekretär	0.7262
6	Finanzminister	0.7172
7	Arbeitsminister	0.7041
8	Ressortchef	0.7022
9	Ministerpräsident	0.7019
10	Außenminister	0.7006

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Partei	0.8445
2	Sozialdemokrat	0.8428
3	Linke	0.8128
4	Liberale	0.7991
5	Regierungspartei	0.7977
6	Koalitionspartner	0.7977
7	Linkspartei	0.7811
8	Parteichef	0.7769
9	Koalition	0.7724
10	Volkspartei	0.7723

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Premier	0.8648
2	Premierminister	0.8314
3	Staatspräsident	0.8031
4	Ministerpräsident	0.7971
5	Oppositionsführer	0.7681
6	Staatschef	0.7583
7	Parlamentspräsident	0.7354
8	Parteichef	0.7330
9	Regierungschef	0.7226
10	Außenminister	0.7189

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	SPD	0.8914
2	FDP	0.8665
3	groß_Koalition	0.8367
4	CDU	0.8148
5	Grüne	0.8115
6	CDU_CSU	0.7994
7	Union	0.7832
8	Rot-Grün	0.7822
9	schwarz-gelb_Koalition	0.7759
10	Schwarz-Gelb	0.7728

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Ständerat	0.8951
2	Nationalrat	0.8601
3	klein_Kammer	0.8117
4	Herbstsession	0.7561
5	vorberatend_Kommission	0.7453
6	Sommersession	0.7399
7	Rechtskommission	0.7388
8	eidgenössisch_Rat	0.7357
9	als_Erstrat	0.7211
10	Wintersession	0.7131

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Parlament	0.8693
2	Abgeordnete	0.8289
3	Parlamentarier	0.7614
4	Volksvertretung	0.7140
5	Volksvertreter	0.7136
6	Nationalversammlung	0.6985
7	beide_Kammer	0.6889
8	Abgeordnetenhaus	0.6875
9	Unterhaus	0.6784
10	Fraktion	0.6689

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Kantonsrat	0.8327
2	Regierungsrat	0.7936
3	Kantonsparlament	0.7743
4	bürgerlich_Mehrheit	0.7245
5	SVP-Fraktion	0.7036
6	SP	0.6999
7	Stadtrat	0.6933
8	Gros_Rat	0.6860
9	Zürcher_Kantonsrat	0.6827
10	Stadtparlament	0.6819

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Vizepremier	0.7767
2	Premier	0.7269
3	Premierminister	0.7227
4	stellvertretend_Ministerpräsident	0.7215
5	Vizeregierungschef	0.6915
6	Minister	0.6767
7	Ministerpräsident	0.6617
8	Verteidigungsminister	0.6536
9	Parlamentspräsident	0.6462
10	Staatspräsident	0.6451

TABLE 6.22: 10 most similar terms to the 10 centroids of the cluster model for the German lexicon for *Representation* Frames in the semantic space of the word2vec model, ordered by cosine similarity

Overall, we see that the concrete governmental bodies of different countries are aptly represented, as well as the political functions of the different political systems[34].

When we turn to Table 6.23 and examine the clusters for the *Efficacy* frames, we find that in contrast to the *Representation* frame, we have much fewer concrete instances or entities present near the centroids. This is no surprise, since the conceptualization of that frame consists of the generic evaluation of results from the implementation of policies, and specifically with an emphasis on the results in contrast to the evaluation of the process (*Efficiency* frame).

Consequently, in centroid 1, we find adjectives which are evaluative and especially apt to describe results. Most of them are negative. Closely linked are centroids 6 and 9, which also reflect negative outcomes but are more typically realized as nouns. Centroid 6 is more dramatical than centroid 9 considering the severity or intensity of the negative outcome. We also find ways to point to unfortunate attempts in centroid 3, which covers more verbs.

On the other hand, more on the positive side, we find terms for victorious and fortunate outcomes in centroid 4. In centroid 8 we find a collection of verbs which consider improvement in general. Also this is a rather positive cluster.

In between the positive and negative evaluation, there is also room for outcomes that just sufficiently meet the requirements and expectations. Terms relating to these middle-grounds are found in centroid 7.

Centroid 2 is about the fulfillment of contractual contents or regulatory laws. In centroid 5, the importance and symbolic intensity of signing contracts and their ratification is reflected. The contents of centroid 10 seem to be a bit less clearly defined. On the one hand, we have verbs that point to successful outcomes (*schaffen, gelingen, erreichen, gewinnen*), on the other hand, we also have terms like *doch* and *eben* in that cluster for which we suggest that they are syntagmatically closely related to evaluations of outcomes (e.g. *da hat eben doch etwas gefehlt* which could be roughly translated to *something was missing after all*). In fact, when we look at a bit further in the environment of the centroid, we find even more "relativizing modifiers" like *aber*

[34]For the calculation of the centroids and their visualization via the vicinity of the centroids, we used a different embedding which is not based on the ten years of news coverage from the three Swiss newspapers which we used throughout all other experiments. For the framing detection and the related experiments we used an embedding based on all (German) texts we collected for the NCCR Democracy project. More precisely, we gathered 547.533 texts containing roughly 300 million words. The preprocessing—lemmatization and removal of punctuation—and the parametrization were the same as for the embedding used for all other experiments. The rationale behind using a different embedding model here is to mitigate the bias towards the Swiss content and include more influence from German newspapers. However, since the Swiss system is still overrepresented in the data sources and therefore in the data collection as well, we still have a slight bias. Additionally, the peculiarities of the Swiss political system also contribute to the resulting "extra Swiss cluster".

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	ungenügend	0.7666
2	mangelhaft	0.7592
3	unzureichend	0.7355
4	unbefriedigend	0.6815
5	miserabel	0.6776
6	schlecht	0.6417
7	unzulänglich	0.6337
8	dürftig	0.6271
9	zufriedenstellend	0.6188
10	lückenhaft	0.6047

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	erfüllen	0.7409
2	umsetzen	0.7104
3	einhalten	0.6999
4	Umsetzung	0.6849
5	Erfüllung	0.6532
6	nachkommen	0.6530
7	Vorgabe	0.6521
8	verbindlich	0.6139
9	Einhaltung	0.5908
10	festlegen	0.5888

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	gescheitert	0.7447
2	scheitern	0.6969
3	missglückt	0.6737
4	misslingen	0.6674
5	Versuch	0.6499
6	Fehlschlag	0.6394
7	Scheit	0.6209
8	glücken	0.6134
9	geglückt	0.6064
10	misslingt	0.6062

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Sieg	0.7728
2	Triumph	0.7092
3	Niederlage	0.7062
4	siegen	0.6936
5	antreten	0.6858
6	Sieger	0.6601
7	besiegen	0.6510
8	triumphieren	0.6494
9	gewinnen	0.6492
10	erringen	0.6413

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	unterzeichnen	0.7968
2	ratifizieren	0.7910
3	Abkomme	0.7883
4	Ratifizierung	0.7858
5	Unterzeichnung	0.7726
6	Vereinbarung	0.7298
7	Ratifikation	0.7071
8	Vertragswerk	0.7044
9	unterzeichnet	0.6930
10	unterschreiben	0.6924

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Desaster	0.8244
2	Katastrophe	0.7254
3	Debakel	0.7202
4	Versagen	0.7132
5	verheerend	0.6689
6	Fiasko	0.6630
7	katastrophal	0.6623
8	Fehler	0.6496
9	Fehlentscheidung	0.6458
10	Misere	0.6322

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	ausreichend	0.8015
2	notwendig	0.7769
3	nötig	0.7652
4	genügend	0.7442
5	vorhanden	0.7435
6	erforderlich	0.7348
7	geeignet	0.7070
8	benötigen	0.7017
9	ausreichen	0.6628
10	eignen	0.6581

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	steigern	0.7547
2	erzielen	0.6594
3	verbessern	0.6541
4	Steigerung	0.6424
5	dank	0.6144
6	übertreffen	0.5983
7	zulegen	0.5979
8	erhöhen	0.5955
9	verbessert	0.5951
10	erreichen	0.5941

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Niederlage	0.7659
2	Rückschlag	0.7352
3	Schlappe	0.7094
4	Scheit	0.6470
5	Enttäuschung	0.6431
6	Debakel	0.6425
7	herb_Niederlage	0.6383
8	schwer_Niederlage	0.6248
9	herb_Rückschlag	0.6230
10	Desaster	0.6117

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	schaffen	0.7964
2	gelingen	0.7764
3	brauchen	0.7586
4	helfen	0.7413
5	doch	0.7380
6	erreichen	0.7272
7	bringen	0.7267
8	fehlen	0.7212
9	gewinnen	0.7210
10	eben	0.7188

TABLE 6.23: 10 most similar terms to the 10 centroids of the cluster model for the German lexicon for *Efficacy* Frames in the semantic space of the word2vec model, ordered by cosine similarity

(but), auch (also), also (anyway/so), zwar (admittely), wohl(arguably/perhaps), gerade(currently), tatsächlich (indeed), vielleicht (maybe), schliesslich (finally), jedenfalls (anyway), eigentlich(actually), obwohl(though) (all with a similarity to the centroid between 0.67 and 0.71). These terms might also point to situations in the texts, where the outcome is not yet fully perceivable, or at least the proper evaluation of the outcome is not obvious enough. However, this centroid represents an interesting mixture of terms for accomplishing a goal and modifying particles that are hedging the original statement.

We need to bear in mind that we see here another instance of a concept which comprises of terms which are antonyms of each other. While the description of success and accomplishment point to the presence of efficacy, the mention of failures and unfortunate attempts to reach a goal are pointing to *absent* efficacy. As both aspects are subsumed in the frame of efficacy, they both contribute to the overall representation via the resource (in the lexicon as well as in the form of specific centroid (i.e., as detector) when we use the clustered re-embedded lexicon).

Finally, we take a closer look at the throughput frame of *Transparency*. Table 6.24 presents the vicinities of the centroids created from the lexicon for *Transparency*.

When we inspect centroid 1, we find an interesting combination that tells a whole story on its own: while it refers to things that are secret (*geheim*) and confidential (*vertraulich*), the same cluster also covers the indiscretion and the disclosure (*Enthüllung*). Additionally, also the leaking of documents (*Dokumente, zuspielen*) and making them public (*publik, publik_machen*) is represented.

The vicinity of centroid 2 mostly consists of verbs of investigation and analysis. This refers nicely to the act of putting something under scrutiny and the reference of investigative effort spent.

Centroid 3 is about verbs of communication. It appears to be linked to the journalists' arsenal of describing utterances in press conferences or interviews. There is also a slight bias to affirmative communication (*erklären, betonen, bestätigen, versichern*) which is a way to communicate if one is defending oneself against allegations—clearly linked to calls for transparency.

There are some rather focused centroids that each capture one specific aspect of transparency: while centroid 4 is about publication in general and making documents public in a narrower sense, we observe the concentrated representation of terms referring to

data in centroid 5. This is certainly also linked to the issue of privacy protection, especially in the realm of online services where big corporate actors like the "Big Four"[35] dominate the field.

In centroid 6, there are terms related to scrutiny and investigative analysis. This centroid is about the concrete (*konkret*) details and particulars (*Einzelheit*) which are checked, discussed and explained (*prüfen, diskutieren, erklären*).

Centroid 7 is about covering up and disguising (*verschleiern, verschweigen, vertuschen, verheimlichen, Verschleierung, kaschieren, unterschlagen*). But it is also linked to debunking (*entlarven*) and reproaching (*vorwerfen*) which are actions that are directly linked to the disclosure of (inappropriate) concealment.

In centroid 8 we find terms which refer to acts of informing and to different ways how information might be gathered. One has to contact (*kontaktieren*) the involved actors and request documents ((*Unterlagen, anfordern*)) or urge them to take a position (*Stellung*$_n$ *ehmen*) or inform about (*darüber_informieren*) something.

Centroid 9 is about transparency itself. Interestingly but not unexpectedly, this cluster contains not only mentions of transparency and comprehensibility, but also its counterparts like intransparency and opacity (*undurchsichtig*).

Finally, centroid 10 is more about clarification and concretion (*darauf_hinweisen, klarmachen, daran_erinnern, klarstellen*) and admitting ((*einsehen, eingestehen, zugeben*).

Overall, multiple aspects of the *Transparency* frame are nicely covered by the detectors. And we also observe that antonyms (like *transparency* and *intransparency*) get represented by the same detector.

But this is not the case for all groups of terms representing specific aspects of the *Transparency* frame. For example, centroid 7 consists of the topic of disguise and concealment. What would be the opposite thereof? Several other centroids cover meanings that are contrary to centroid 7: centroid 4 is about making information publicly available which is changing the state of information that is hidden. Centroid 6 is about the details and the comprehensive checks that also covers partly the opposite of centroid 7 if we interpret the latter more in the sense of a cover-up. And also centroid 8 which is about providing (requested) information is at least contrary to the interpretation of centroid 7 geared towards concealment. Lastly, even centroid 7 contains aspects that reverse the state of hidden information like *debunk* (*entlarven*).

[35] These are the tech companies Google, Amazon, Facebook, and Apple; see `https://en.wikipedia.org/wiki/Big_Four_tech_companies`; "Big Five" would also include Microsoft as well.

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	geheim	0.7114
2	vertraulich	0.7112
3	Indiskretion	0.6511
4	Dokument	0.6499
5	publik	0.6448
6	zuspielen	0.6421
7	enthüllen	0.6378
8	publik_machen	0.6369
9	Enthüllung	0.6239
10	brisant	0.6217

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	dokumentieren	0.7733
2	untersuchen	0.6969
3	analysieren	0.6896
4	aufklären	0.6574
5	aufarbeiten	0.6510
6	auswerten	0.6469
7	detailliert	0.6322
8	zusammentragen	0.6315
9	anhand	0.6276
10	recherchieren	0.6245

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	erklären	0.8100
2	betonen	0.7811
3	bestätigen	0.7794
4	behaupten	0.7448
5	berichten	0.7366
6	versichern	0.7308
7	erfahren	0.7205
8	wissen	0.7110
9	meinen	0.7084
10	sagen	0.7042

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	publizieren	0.7620
2	veröffentlicht	0.7566
3	publiziert	0.7408
4	verfassen	0.7088
5	Publikation	0.6951
6	veröffentlichen	0.6913
7	ausführlich	0.6796
8	herausgeben	0.6566
9	Dokument	0.6529
10	vorliegend	0.6475

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Daten	0.7924
2	persönlich_Daten	0.7126
3	Information	0.7036
4	sensibel_Daten	0.6946
5	Datensatz	0.6822
6	personenbezogen_Daten	0.6669
7	Kundendaten	0.6651
8	Zahlungsinformation	0.6607
9	PNR-Zentralstelle	0.6450
10	gespeichert_Daten	0.6425

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Detail	0.7215
2	vorlegen	0.7167
3	konkret	0.7094
4	detailliert	0.6943
5	prüfen	0.6866
6	diskutieren	0.6850
7	klären	0.6727
8	umfassend	0.6457
9	informieren	0.6452
10	Einzelheit	0.6352

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	verschleiern	0.7321
2	verschweigen	0.7221
3	vertuschen	0.7036
4	verheimlichen	0.6400
5	Verschleierung	0.6116
6	suggerieren	0.6034
7	kaschieren	0.5967
8	unterschlagen	0.5967
9	entlarven	0.5961
10	vorwerfen	0.5927

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	informieren	0.7243
2	Auskunft	0.6943
3	kontaktieren	0.6557
4	Unterlage	0.6283
5	anfordern	0.6271
6	darüber_informieren	0.6255
7	Stellung_nehmen	0.6111
8	weiterleiten	0.5974
9	abklären	0.5891
10	erkundigen	0.5867

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	transparent	0.7702
2	Transparenz	0.7146
3	offenlegen	0.6708
4	intransparent	0.6554
5	mehr_Transparenz	0.6250
6	nachvollziehbar	0.6237
7	Intransparenz	0.6045
8	Offenlegung	0.5977
9	undurchsichtig	0.5889
10	objektiv	0.5829

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	darauf_hinweisen	0.7410
2	einsehen	0.6940
3	eingestehen	0.6824
4	klarmachen	0.6780
5	daran_erinnern	0.6484
6	Eindruck_erwecken	0.6484
7	darüber_informieren	0.6468
8	klarstellen	0.6419
9	davon_ausgehen	0.6214
10	zugeben	0.6046

TABLE 6.24: 10 most similar terms to the 10 centroids of the cluster model for the German lexicon for *Transparency* Frames in the semantic space of the word2vec model, ordered by cosine similarity

6.5 Chapter Summary

In this chapter we have presented a variety of empirical results from experiments considering lexicon induction. We started with a thoroughly detailed description of the simple case of dog breeds as an introductory example. This was also an example to demonstrate the simplest case for lexicon induction where we just intend to enlarge a given set of entities, or, in other words, to find "more of the same". This focuses especially on cases where we are looking for new entries which are of the same kind and not only related to the original set. In our example, we just want to find other dog breeds and filter out candidate terms that are related in some way but are not dog breeds themselves.

We then shed a light on two combinatory cases: on the one hand, we showed how to combine concepts such as "verbs of communication" with "negative sentiment" and conveniently create new resources, thus avoiding large manual efforts. For this case we also referred to methods that foster the induction process by injection of promising candidate triples that we gather from an intermediate clustering step.

On the other hand, we depicted how we adapt a rather generic concept such as crime to a specific economic sector. Although we started with a really small generic concept—crime was defined by only 11 terms—we have shown how to easily derive such a resource. For this scenario of expansion combined with domain adaptation, we also proposed another way to set up the lexicon induction in a largely unsupervised fashion.

More concretely, we argued for multiple passes of the scripted induction in order to profit from the mitigation of erroneous entries via an aggregation step before re-embedding the resource. Alternatively, the aggregation of results from the multiple passes may also be interpreted as probabilistic corroboration in the sense that terms with the highest aggregation scores were induced "from multiple angles" (remember the degree of randomness that is included in the algorithm). In contrast, the errors are vanishing since their erroneous induction (caused by inapt random sampling in the unsupervised search procedure) is not systematic.

In a fourth case, we reported the results of deriving a sentiment lexicon from a small seed lexicon. We provided this case because it allows for a quantitative evaluation since we had a large sentiment lexicon already available. Additionally, we manually evaluated the induction of new negative terms in order to estimate the quality of the newly added terms which are not yet included in the sentiment lexicon. Also in this case the findings support the assumption that the aggregation of multiple runs is usable as a proxy to estimate the confidence for the new resource that was generated in an unsupervised fashion. Additionally, this case demonstrates how separated antonymous resources may be created although they are heavily intertwined in the basic resource, the embedding.

Subsequently, we have turned to the case of document classification—especially to small data sets with a skewed distribution. For this case, we have argued that we seek to address the problems that arise in standard machine learning approaches by externalizing the generalization that we strive to achieve.

We therefore proposed to induce a lexicon for each class we want to detect and demonstrated in this scenario how we make use of the little data that is available and combine it with the productive lexicon induction procedure.

Furthermore, we also discussed in detail the detectors that we calculate with the induced lexicons for one of the three document classification cases. This is done with two goals in mind. Firstly, we think the view on the resource via the detectors is the most insightful, given the downstream application of document classification. It provides a concise overview of the captured overarching concept for the class. Secondly, while the projection of the induced lexicon into detectors is done partly for pragmatic reasons (see Chapter 5), we additionally get semantically motivated sub-concepts for each lexicon to which we may easily refer back to when applying these resources for classification. We will evaluate the usefulness of the generated resources quantitatively in the following chapter where we apply a comparative perspective to benchmark different approaches for document classification, including the proposal from this work.

Finally, in a similar way, we reviewed the resources which we create in order to tackle the challenge of framing detection (see Chapter 8). After a short introduction to the framing approach, we presented the lexical resources for three out of ten framing categories, again through the inspection of the clustered re-embedded version of the lexicon.

For each of the coarse-grained categories of the legitimacy frames, we present one example of the fine-grained categories subsumed by them (input: representation, output: efficacy, and throughput: transparency). While the lexicon induction process is similar to the one used for document classification, the actual task of framing detection poses several additional challenges. Especially the strongly locally bounded evidence and a disproportionally large residual class make this task hard.

To sum up, this first chapter on the empirical outcomes is meant to demonstrate the versatility in application and purpose of the proposed lexicon induction procedure. It also lays the foundation for the following two chapters devoted to document classification and framing analysis.

7

Experiments II: Lexicon-based Document Classification

> "Supposing is good, but finding out is better."
> — Mark Twain

```
In [31]: analogy(a="Experiment", b= "Prüfstein",
x="Messung", y=None, model_given=model, verbose=True)
'Experiment' is to 'Prüfstein' as 'Messung' is to 'Nagelprobe'
Out[151]:
[('Nagelprobe', 0.3974507451057434),
 ('Gradmesser', 0.37944966554641724),
 ('Indikator', 0.3719639778137207),
 ('Messstation', 0.3604757785797119),
 ('Belastungsgrenzwert', 0.35998445749282837),
 ('Härtetest', 0.35921812057495117),
 ('Messwert', 0.3484167158603668),
 ('Grenzwert', 0.34523534774780273),
 ('massiv_Überschreitung', 0.34500551223754883),
 ('Tagesmittel', 0.3411290943622589)]
```

In this chapter, we apply our approach to the task of text classification. More precisely, we tackle a specific setup which is challenging for standard machine learning approaches

to text classification. On the one hand, we will work with data sets that incorporate a large skew in the distribution between the given classes. On the other hand, the data sets used in this chapter are of noticeably smaller size (especially compared to the evaluation data sets used in computational linguistics for this task like the 20 newsgroup data set[1], consisting of roughly 20.000 data points or data sets from newer research used in Zhang et al. (2015), Joulin et al. (2017), or Shen et al. (2018) where the number of labeled data points is in the hundreds of thousands).

For data-driven methods, such settings normally pose a challenge, because generalization is harder with less training data, and simple measurements to counteract the imbalance of data (down-sampling and up-sampling) are not applicable without aggravation of the sparse data problem. However, such settings are rather common in social science where a data set comprising of hundreds of documents is not considered small and data sets of tens of thousands of examples is rare to say the least.

While tackling such a widely prevalent problem may illustrate the usefulness and versatility of the approach, we also argue that the experimental results allow for clearer investigation of the strengths and weaknesses of the approach.

7.1 Tackling the Skewed Data Distribution Problem

Skewness in the data distributions always poses a challenge to the generalization of learning algorithms. This is especially linked to the cases where the underrepresented classes do not provide us with a sufficient amount of examples so that a generalization is properly learned. In simpler terms: There are not enough examples for the classifier to build a model which generalizes over the given classes.

While there are numerous propositions how to counteract skewness of data distributions from the perspective of the classification task itself (e.g., over-/undersampling), they generally do not focus on the generalization aspect of the problem.

The method we propose in this work is to externalize the generalization as a (concept) lexicon inducing task. By following this path we try to leverage several possible advantages:

- First—and most obviously—we make use of the tools for lexical induction[2] presented here beforehand.

[1] see `http://qwone.com/~jason/20Newsgroups/`

[2] Of course, we could also start with a given lexicon if available and improve or adapt it in the lexicon induction process.

- Second, since we rely strongly on a generic semantic model, we are able to profit from the semantics represented in this (also external) model. In other words: we do not need to learn the correlations and distinctiveness of the vocabulary considering the target classes from scratch (or better: based on the given data for the classification). Rather, we use the meaning contained in distilled (and re-embedded) lexicons that serve as the basis for the decision. To further profit from the generalization of the distributional semantics, we apply an embedded modeling described in Chapter 5 to allow for generalization during prediction.

- Third, because we externalize the resource generation, we are able to create an (approximately) equivalently generalized representation also for the small classes, i.e., for those classes for which only a few examples are available in the collection. This is also reflected in the number of detectors which is set to be equal for all classes and therefore independent of the number of provided examples in the data set.

Classes as Concepts

We have shown in the preceding chapter how to derive the concept lexicons for generic classes like positivity or negativity in the context of sentiment analysis, and we have also used a similar technique to derive more specific concept lexicons. We briefly recap here the central points from the induction process and posit the way we derived the seed lexicons under the perspective of the comparability.

As mentioned before, the goal was to derive a concept lexicon for each class that we shall be able to predict in the text classification task. For reasons of comparability, we induced a core of terms from the same data split which is used in the other data-driven approaches, i.e., we use the static 80/20 stratified data set split to generate a set of important terms for each class[3].

In a second stage, we use the tools described in the previous chapter to expand the lexicons. An overview of the derived cores and expanded concept lexicons is found in Table 7.1, 7.2, and 7.3.

[3]For this first step we used the `SeedFinder` component of the `ABCD` package which applies a mixture of statistical tests to extract words from labeled data which occur uncommonly often. This is calculated in relation to the other classes, the corpus as a whole, and to an external general word frequency table, retrieved from the package `wordfreq` (Speer et al., 2018).

However, any filtering process that identifies important words in texts is applicable. Furthermore, the seed does not need to be encompassing or exact (in the sense of disjunctivity) for the lexicon induction process which in turn also does not have to adhere to such criteria. This is mainly due to the properties of the re-embedding step of the lexical resources described in Chapter 5 and 6.

While the goal is to enable the incorporation of as much a priori knowledge as possible (be it conceptual or in the form of already labeled data), these lax constraints for the seed and the lexicon may be seen as layers of robustness from which the downstream performance of the resources profit.

Subcategories	**Words in Seed**	**Words in Lexicon**
Beruf/Berufsbildung	39	257
Bildung/Schule/Hochschule	112	407
Wissenschaft/Forschung/Technik	124	305

TABLE 7.1: Number of words in the lexicon and the seed set to represent the subcategories for the domain *Bildung (Education)*

Subcategories	**Words in Seed**	**Words in Lexicon**
Abfall	16	178
Klima	177	291
Natur/Landschaft	78	230
Raumplanung	20	238
Tiere	133	425

TABLE 7.2: Number of words in the lexicon and the seed set to represent the subcategories for the domain *Umwelt (Environment)*

Subcategories	**Words in Seed**	**Words in Lexicon**
Güterverkehr	28	123
Luftverkehr	95	441
Raumfahrt	61	225
Schienenverkehr/Bahn	82	275
Schifffahrt	17	182
Strassenverkehr	46	460

TABLE 7.3: Number of words in the lexicon and the seed set to represent the subcategories for the domain *Verkehr (Traffic)*

7.2 Setting for the Experiments

In this section, we briefly describe the approaches that are compared in the classification experiments. We also motivate the rationale of the comparison with our proposed approach.

In the experiments, we compare our approach to the following contestants:

- The baseline is a "classic" machine learning approach using an SVM (Support Vector Machine) algorithm based on a TF-IDF weighting of the input. This approach performs well for a wide range of text categorization and is chosen to represent

models that are purely data-driven and relying on a bag-of-word modeling. We report results on two versions of this baseline. First, we report the (averaged) performance for the classifier from a 5-fold stratified cross-validation ("Baseline CV 5"). Second, we also report the results on a fixed 80/20 split of the data sets ("Baseline 80/20"). This is the split that we will also use for the other approaches.

- A next point for comparison is a purely dictionary-based and somewhat simplistic classifier. More concretely, we use the same lexical resources we have created, but instead of the more sophisticated modeling in the embedded space, we simply count the words of the different lexicons in the article and use these counts as features for a classifier ("Lexicon based"). Logistic regression ("LR") and a Random Forest ("RF") classifiers are tested. We report the setting with the best performance.

- Naturally, we report the results for the proposed classifier with the default settings of our approach ("ABCD" which refers to "all based on concept detectors").

- Additionally, we compare here also with a slightly tuned version of the classifier in the sense that we report the best results from a grid search on the two parameters of the classifier in the given case at hand ("ABCD tuned").

- In order to set up a touchstone, we also train a classifier that is partly geared to solve such scenarios, namely `fastText` (Joulin et al., 2017), once without a given pre-trained embedding, once given a pre-trained embedding (based on all the same raw articles as we have used to derive the `word2vec` model underlying the proposed approach).

  We thus apply the `fastText` classifier in three variants: for the first two variants, we evaluate using the same 80/20 split from the data, which is also used for the baseline, as well as for the lexicon-based classifier. We include a version without pre-trained vectors ("fastText NPV 80/20") and one with a given pre-trained embedding ("fastText 80/20"). To illustrate especially the influence of the given labeled data, for the third variant we report results for the inverse split, i.e, we use the 20 percent of the data (which was used for testing before) for the training of the classifier and 80 percent for testing ("fastText 20/80"). This means that the classifier has to face a larger variance during prediction while shown less variance during training in comparison to the 80/20 split. We hence illustrate the deterioration of the performance given the decrease of training data.

The comparison with a reasonable baseline explains the benefits for the specific problem at hand, namely the skew in the data distribution between the classes. This is one of the main declared goals of this thesis: to improve classification on small data sets incorporating a large skew in class distribution. Especially the smallest class is often

not represented well in the model and therefore suffers from poor recall. To investigate this problematic outcome and the result of the efforts to counteract this, we will also compare confusion matrices of the baseline classifier and the ABCD approach.

The reason why we present the performance of this baseline classifier in two variants is to illustrate the difference in performance between the averaged 5-fold cross-validation and the static 80/20 test/train split we use respectively. In other words, the discrepancy between the result of the static split and the cross-validation serves as a hint to estimate if the drawn static split was particularly easy to predict or especially hard[4].

The cross-validation measurement is taken for the baseline classifier, but not for the approaches relying on the lexical resources. The reason for this is that it was not possible not start five times with a different lexicon induction from scratch because of the involvement of the researcher in the interactive lexicon generation process. More precisely, since we have already seen the result of the core lexicon and its extension from the first run based on the static sample, it would have been nearly impossible to repeat this process with one of the (new) cross-folds without unintentionally drawing from this knowledge[5]. To sum up, we hold all preconditions equal between the baseline and the ABCD approaches, but for the baseline we also report results based on a cross-validation instead of the static split used otherwise.

The reports on the classifier that is purely relying on the lexical resources serves as a point of reference to estimate how much of the performance from the proposed ABCD approach is caused by its different modeling approach. In more detail, the lexicon-based classifier on the one hand uses features that are document-wide counts from lexicon look-ups. While it profits on the generalization gained from the lexicon induction process, there is no sophisticated sentence-level modeling for the feature generation. The ABCD approach on the other hand applies a sentence-level prediction based on an embedded modeling (of the input and the lexical resources). Therefore, the difference in performance between this point of reference and the ABCD approach lets us observe the consequences of the different modeling approaches.

[4] Additionally, we would also argue that the difference between these two measurements may also be used to reason about the robustness of such approaches. While it is true that the purely computational feasibility of such approaches has reached a level that allows for extensive parameter tuning (e.g., by carrying out a grid search over the parameters), the robustness can be estimated and confronted with the variance in performance given different samples. If we observe large differences in performance between different sampling rounds, then the robustness (or in other terms: the achieved generalization) should be questioned. This holds true, of course, for differences between the folds and between the average of the cross-validation and the static split since the latter may also be seen as an additional fold.

[5] To counteract this flaw, one would need to conduct the experiment with n humans independently who weakly supervise the lexicon induction for n folds, given the according data splits. Of course, this introduces another methodological problem in the form of variance between the behaviour of the involved humans.

The performance of the "ABCD tuned" version is reported in order to show the possible potential of optimization through the choice of the two parameters (threshold and best n).

Finally, the comparison with the `fastText` classifier is used to measure the competitiveness of our approach. While we have an additional data point to estimate the potential of embedded approaches through the comparison of the fastText NPV 80/20 and the fastText 80/20 (the latter one makes use of a large embedding), we also attempt to compare the performance of a data-driven embedded approach (fastText 80/20) and a concept-driven embedded approach (ABCD). For the sake of comparability, we also preprocess the data in the same manner as it is inherent for the ABCD approaches in the sense that we lemmatize and filter for nouns and adjectives[6].

Last but not least, there is another difference between the presented contestants. It has to be mentioned again that all the classifiers beside the ABCD use labeled data to train while the proposed approach in its pure form is actually a heuristic (or rule-based) classifier and does not need annotated data. The only step that (partially) requires supervision is the lexicon generation process (see Chapter 4 and Chapter 6).

7.3 Description of the Task

For this experiment we use a data set which is sampled from a corpus of labeled articles from Swiss newspapers. The articles in this corpus are labeled manually for archiving and indexing purposes[7]. We consider those labels further as gold standard against which we compare out method. For the sake of simplicity, we remain in a monolingual scenario, using only German articles. Additionally, although originally stemming from a multi-label classification corpus, we restrict the sampling to single-labeled articles, again for the sake of simplicity and, alongside, in order not to induce additional difficulty for standard text classification approaches.

The articles of the corpus are categorized into nine top-level categories and on a second level between several subcategories per top-level category. The task is to predict the subcategory of an article (given that all articles in the data set are from the same top-level category).

[6] We also compared the performance on raw text for the `fastText` classifier, i.e., without lemmatizing and filtering but instead trying to leverage word n-grams. However, the results of the preprocessed versions are better throughout the three domains. Therefore we report only those.

[7] This is carried out by the Schweizer Mediendatenbank

7.3.1 Measurement

In the following section presenting the results we will report *Precision*, *Recall*, *F1-score* and *Accuracy*. Since we are especially interested in the ability to cope with skewed data distributions, we will report macro-averaged scores as well, in order to reflect the influence on the classes with few examples (see also Chapter 4 in Jurafsky and Martin (2019)).

Additionally, we also give a detailed evaluation for all classification tasks in the form of a confusion matrix including the results from the baseline and the ABCD approach.

7.3.2 Data Description

For the experiment we use articles from the Swiss Newspapers "Tages-Anzeiger" (TA), "Neue Zürcher Zeitung" (NZZ), "Blick" (BLI) from the years 2004 to 2006. These outlets are the biggest in the German-speaking part of Switzerland and cover a wide range of topics. While the NZZ is a well-established quality-paper of international reputation (broadsheet), the TA has a slightly more regional focus on Zürich and the Blick is the largest national outlet of the yellow press (tabloid).

We take the class labels (which are manually assigned by the Schweizer Mediendatenbank) that we obtained with the documents as gold standard. In the following, we briefly describe the three data sets used for the evaluation and visualize the skewed data distribution.

The first set used in the evaluation contains texts about education (n=345). The subcategories for education which the algorithms strive to learn to label are *Bildung/Schule/Hochschule* (*Education/School/University*), *Beruf/Berufsbildung* (*Professions/Vocational training*), and *Wissenschaft/Forschung/Technologie* (*Science/Research/Technology*).

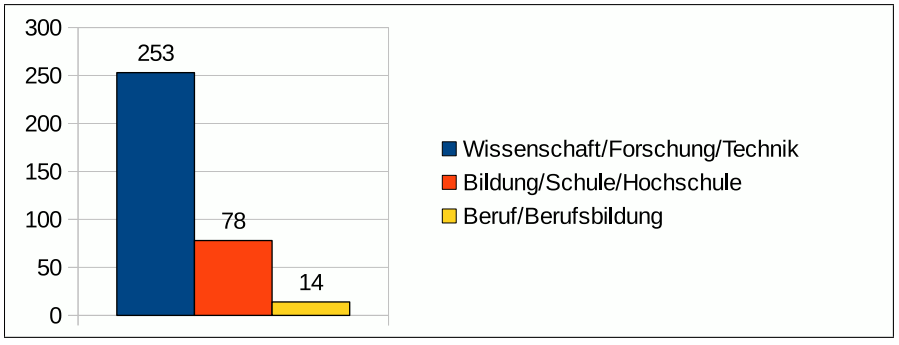


FIGURE 7.1: Distribution of classes over texts in the domain *Bildung (Education)* in absolute counts

The data set has a large skewness in the class distribution, which is observable in Figure 7.1. The ratio between the largest and the smallest class is 18.07 (73.3% vs. 4.1%). Considering the content and the given labels, the distinction between *Education/School/University* and *Science/Research/Technology* becomes difficult in those articles about science and research or about an institution of higher education where a University context appears in both cases.

The second set used in the evaluation contains texts about the environment and nature (n=327). The subcategories to identify are *Tiere* (*Animals*), *Klima* (*Climate*), *Natur und Landschaft* (*Nature and Landscape*), *Raumplanung* (*Spatial Planning*), and *Abfall* (*Waste*).

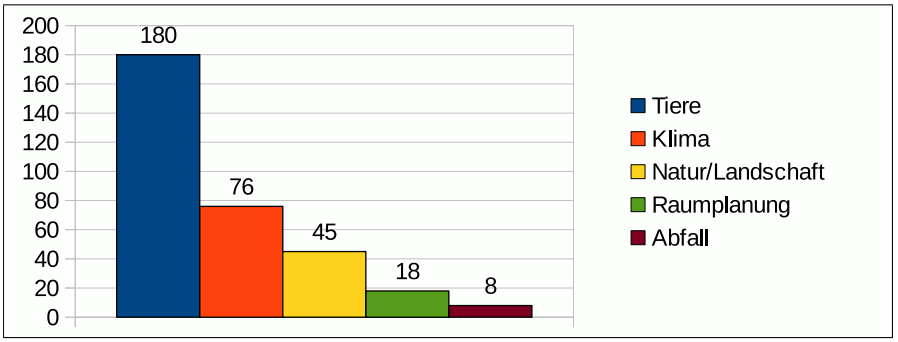


FIGURE 7.2: Distribution of classes over texts in the domain *Umwelt (Environment)* in absolute counts

In this categorization we have the two rather broad categories *Animals* and *Nature and Landscape*, which partly overlap considering the underlying concept (as well as with the more narrowly defined other classes). However, the goal is to predict the most prevalent perspective or aspect of an article. Nevertheless, we will have to reflect this finding in the discussion of the results below.

Additionally, Figure 7.2 shows that the data set contains only a few examples for the categories *Abfall* and *Raumplanung*, which makes it challenging to generalize these concepts for purely data-driven classifiers, as the given example data is scarce. The ratio between the largest and the smallest class is 22.5 (55.1% vs. 2.5%).

The third set used for evaluation purposes contains texts about traffic (n=321). The subcategories for education which to distinguish from each other are *Luftverkehr* (*Air Traffic*), *Schienenverkehr/Bahn* (*Railway Transport/Train*), *Strassenverkehr* (*Road Traffic*), *Raumfahrt* (*Space Travel*), *Schifffahrt* (*Shipping*), and *Güterverkehr* (*Freights Traffic*).

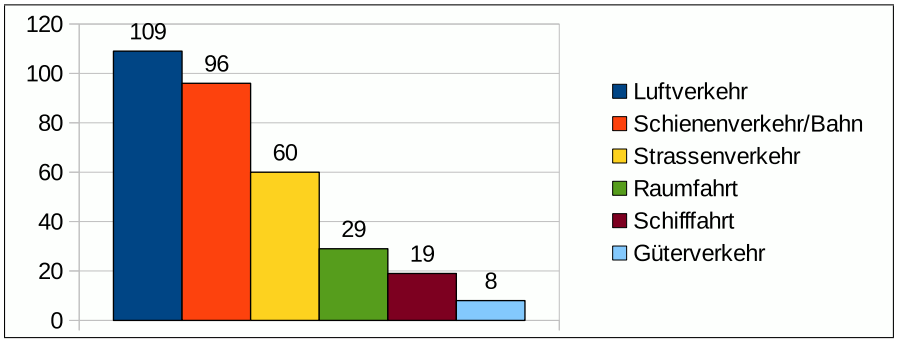


FIGURE 7.3: Distribution of classes over texts in the domain *Verkehr (Traffic)* in absolute counts

Again, we have a set of categories which differ in specificity: while *Air Traffic*, *Space Travel*, and *Shipping* are clearly distinguishable, *Railway Transport/Train* and *Road Traffic* overlap in those cases where both appear in an article (e.g. discussion about which way of transport is preferable). *Freights Traffic*, on the other hand, may occur together with all means of transportation, so again, the main aspect of the article is the heuristic to determine the class. In addition to this, we see that, similarly to the five subcategories of the *Umwelt* domain, there is a skew in the class distribution over the six subcategories (cf. Figure 7.3). The ratio between the largest and the smallest class is 13.6 (34.0% vs. 2.5%).

In the following sections we report the classification performance for all described classification systems considering one of the aforementioned data sets.

7.4 Results

In this section we briefly report the results in the form of a quantitative evaluation. We point to remarkable differences as well as unexpected outcomes and also give detailed insights for specific cases. We especially investigate the difference of the proposed approach compared with the baseline and try to address the given rationale of the set of contestants (cf. section 7.2).

7.4.1 Results for the Classification in the Domain *Bildung (Education)*

If we compare the results for the performance of the different classifiers given in Table 7.4, we easily observe that they all reach a high accuracy (over 0.9). But this measurement does not reflect the imbalance of quality for the prediction per class. If we focus on

macro measurements[8], we see that the baseline suffers from low recall and precision, besides the macro precision from the Baseline CV5.

Similarly, the lexicon-based approach and the `fastText` classifier cannot deliver a recall which matches the micro average level. This in turn means that at least one class is not predicted well.

	P_{micro}	P_{macro}	R_{micro}	R_{macro}	$F1_{micro}$	$F1_{macro}$	Acc
Baseline CV5	0.91	0.94	0.91	0.63	0.91	0.68	0.91
Baseline 80/20	0.91	0.63	0.91	0.60	0.91	0.61	0.91
Lexicon based (LR)	0.91	0.95	0.91	0.71	0.91	0.77	0.91
ABCD	0.96	0.95	0.96	0.97	0.96	0.96	0.96
ABCD tuned	0.96	0.97	0.97	0.97	0.97	0.97	0.97
fastText NPV 80/20	0.94	0.98	0.94	0.74	0.94	0.80	0.94
fastText 80/20	0.94	0.95	0.94	0.75	0.94	0.79	0.94
fastText 20/80	0.90	0.90	0.90	0.62	0.90	0.63	0.90

TABLE 7.4: Evaluation of Classification in the Domain *Bildung (Education)*

When we turn to the detailed evaluation in the form of a confusion matrix and compare the baseline with the given approach, it becomes clearer why the results of macro and micro averages diverge. As we see in Figure 7.4, the baseline classifier completely fails to recognize the class *Beruf/Berufsbildung*. In contrast, the same class (although containing only a few instances) is correctly detected by the ABCD approach, as visible in Figure 7.5. In fact, only two articles are misclassified by the ABCD approach.

By further scrutinizing those two articles in detail, we find out that one of the two misclassified articles is about Albert Einstein's career in primary school. Although this article should be classified as member of the class *Wissenschaft/Forschung/Technologie*, it is clear why the classifier erroneously predicted the wrong label (*Education/School/University*).

The other misclassified article contains a review of a book on the historical development of the *ETH Zürich (Swiss Federal Institute of Technology)*. The article discusses several aspects of the *ETH* as an academic institution as well as an instance of a school for higher education.

[8]For macro averages, the performance for each class is taken into account with the same weight. This puts the same emphasis on the smaller classes, due to the identical contribution to the final measurement. This leads arguably to a distorted picture if one is interested in labeling as many single cases correctly as possible. If we instead focus on equal quality in performance for all classes, the macro average reflects the optimization at the cost of smaller class well. Complementarily, we report also micro-averaged numbers in the table.

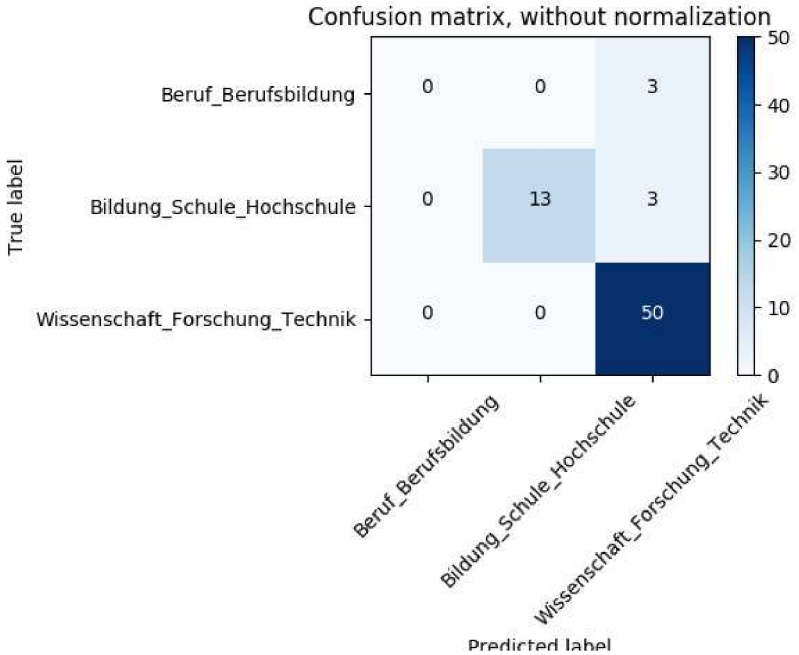


FIGURE 7.4: Confusion matrix for predictions of the baseline classifier for the domain *Bildung (Education)* in absolute counts

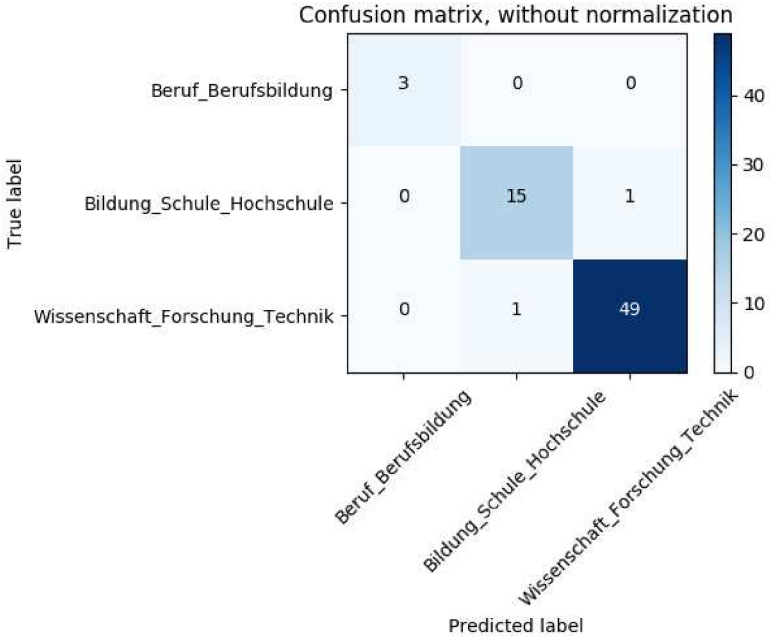


FIGURE 7.5: Confusion matrix for predictions of the ABCD-classifier for the domain *Bildung (Education)* in absolute counts

Due to the strong bias of the *ETH* towards the technological aspects of science mentioned in the article, the classifier gives the label *Wissenschaft/Forschung/Technologie*. While this is the wrong label compared against the gold standard, this decision is arguably not completely wrong.

Besides the promising result in this tri-partite classification, the manual evaluation of the misclassified texts also serves as a hint that the erroneous labeling is a still explicable.

7.4.2 Results for the Classification in the Domain *Umwelt (Environment)*

For the domain *Umwelt* the fastText 80/20 classifier delivers the best performance. We also observe that the baseline is behind all other classifiers, although the performance in terms of micro measurements is not bad.

If we consider the macro measurement and especially the macro F1 score, we notice that the baseline with its generalization based only on data-driven vocabulary-specific weighting, is surpassed by the externally generalized lexical approach. This is linked to the low macro recall, which in turn is an effect of the unsatisfying representation of the smaller classes *Natur* and *Landschaft* as well as *Raumplanung*. In other words, when only few examples for some classes are available, approaches such as the baseline classifier only have a weak grounding for the prediction, due to the thin evidence.

To study this outcome in more detail, we may put the confusion matrices under scrutiny (Figure 7.6). For the baseline classifier we observe a poor prediction quality for *Raumplanung*, which is only predicted correctly for one article[9].

The fact that the lexicon-based approaches perform better in this scenario corroborates the usefulness of external generalization in the form of lexicon induction and expansion as it was applied here (cf. Chapter 6). We also observe that the embedded modeling from the ABCD approach improves the prediction in comparison with the lexicon-based approach. Additionally, the performance is further improved when the parameters are optimized (ABCD tuned).

As mentioned in the beginning, the fastText 80/20 classifier performs best for this task. This shows that there is still room for improvement for the ABCD approach, in contrast to the previous example in the domain of *Bildung* where the only misclassifications would not be correctable while adhering to the same premise, i.e. without changing the modeling (remember the qualitative error analysis for the two articles).

When we turn to the fastText 20/80 model, which relies on only one quarter of the labels in comparison with the fastText 80/20 model, we notice that the decrease in performance

[9]Interestingly the even smaller class *Abfall* was predicted perfectly from all approaches. This may be intuitively explained as its main topic is distinctive enough to be represented well given only few examples

	P_{micro}	P_{macro}	R_{micro}	R_{macro}	$F1_{micro}$	$F1_{macro}$	Acc
Baseline CV5	0.81	0.85	0.81	0.58	0.81	0.65	0.81
Baseline 80/20	0.85	0.88	0.85	0.70	0.85	0.74	0.85
Lexicon based (RF)	0.86	0.86	0.86	0.80	0.86	0.82	0.86
ABCD	0.89	0.90	0.89	0.87	0.89	0.88	0.89
ABCD tuned	0.92	0.93	0.92	0.87	0.92	0.90	0.92
fastText NPV 80/20	0.85	0.68	0.85	0.62	0.85	0.63	0.85
fastText 80/20	0.97	0.96	0.97	0.94	0.97	0.95	0.97
fastText 20/80	0.88	0.78	0.88	0.76	0.88	0.76	0.88

TABLE 7.5: Evaluation of Classification in the Domain *Umwelt (Environment)*

is substantial. This has to be kept in mind when comparing it to the proposed ABCD approach which is not relying on labeled data to train a model for prediction[10].

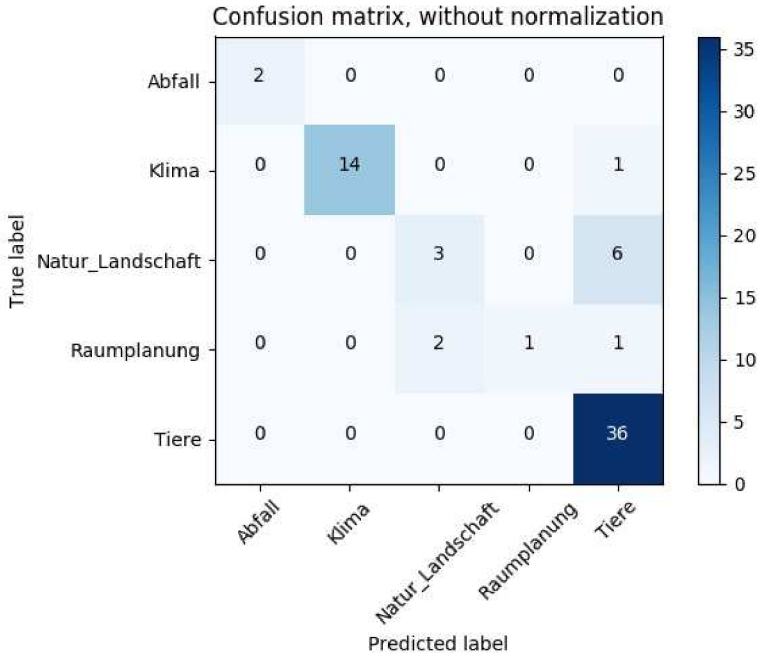


FIGURE 7.6: Confusion matrix of the baseline classifier for the domain *Umwelt (Environment)* in absolute counts

[10]To be more precise, the classifier has no access to the labeled data. However, the lexicon induction process was seeded with a "core" of the concept derived from the original 80/20 split, i.e., based on 80 percent of the labeled data.

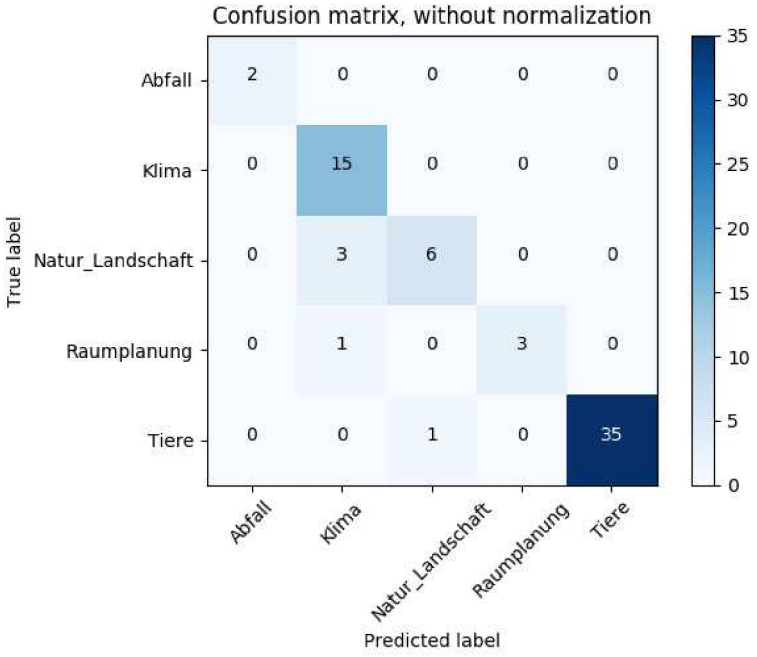


FIGURE 7.7: Confusion matrix for predictions of the ABCD-classifier the domain *Umwelt (Environment)* in absolute counts

7.4.3 Results for the Classification in the Domain *Verkehr (Traffic)*

Also for the domain *Umwelt* the fastText 80/20 classifier reaches the highest scores.

Considering accuracy, we state that all approaches score well (between 0.89 and 0.95 accuracy). If we turn to the macro scores, we see a decrease in performance for the baseline classifiers (F1 macro scores below 0.8) which is mainly due to the macro recall. In this specific scenario, this is caused by the bad performance for *Güterverkehr*, the smallest class[11]. We observe again in the confusion matrices in Figure 7.8 and Figure 7.9 that the baseline classifier fails to provide an apt representation of *Güterverkehr* which is —in terms of recall—sufficiently modeled with the lexicon-based approaches.

Interestingly, in this domain we find that the pure lexicon-based classifier beats the ABCD approaches. This means that in this case the ABCD modeling is not advantageous. While the recall on macro measurement is exactly equivalent and almost on par in micro measurement, the ABCD approach is substantially outperformed by the lexicon-based approach on the precision measurement.

However, again the fastText 80/20 model is the best, scoring even better than the lexicon-based model (0.95 vs. 0.91 in terms of accuracy, 0.97 vs. 0.94 in terms of F1

[11] Of course, macro measurements have a bias to exaggerate the influence of the small classes due to averaging on class-level. However, this is the reason why we present micro and macro scores next to each other.

macro). It is worth mentioning that also the fastText 20/80 model performs well in this scenario, mainly due to the good precision. But also for recall—in contrast to the cases of the domain *Bildung* and *Umwelt*—the drop in performance is not so drastic if we inverse the data for the fastText classifier.

	P_{micro}	P_{macro}	R_{micro}	R_{macro}	$F1_{micro}$	$F1_{macro}$	Acc
Baseline CV5	0.90	0.95	0.90	0.76	0.90	0.79	0.90
Baseline 80/20	0.91	0.79	0.91	0.78	0.91	0.78	0.91
Lexicon based (LR)	0.91	0.93	0.91	0.94	0.91	0.94	0.91
ABCD	0.89	0.82	0.89	0.94	0.89	0.86	0.89
ABCD tuned	0.91	0.86	0.91	0.94	0.91	0.89	0.91
fastText NPV 80/20	0.91	0.73	0.91	0.77	0.91	0.75	0.91
fastText 80/20	0.95	0.97	0.95	0.97	0.95	0.97	0.95
fastText 20/80	0.91	0.94	0.91	0.86	0.91	0.89	0.91

TABLE 7.6: Evaluation of Classification in the Domain *Verkehr (Traffic)*

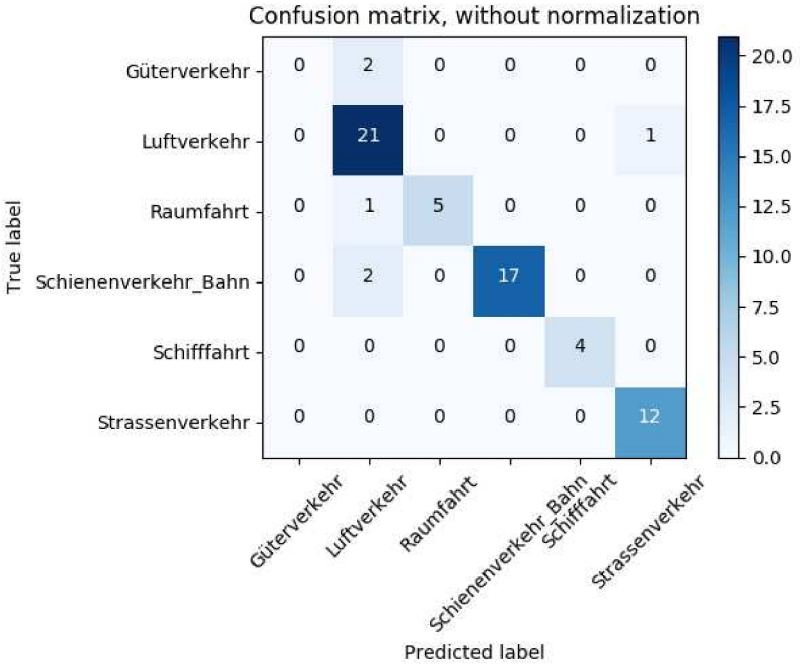


FIGURE 7.8: Confusion matrix of the baseline classifier for the domain *Verkehr (Traffic)* in absolute counts

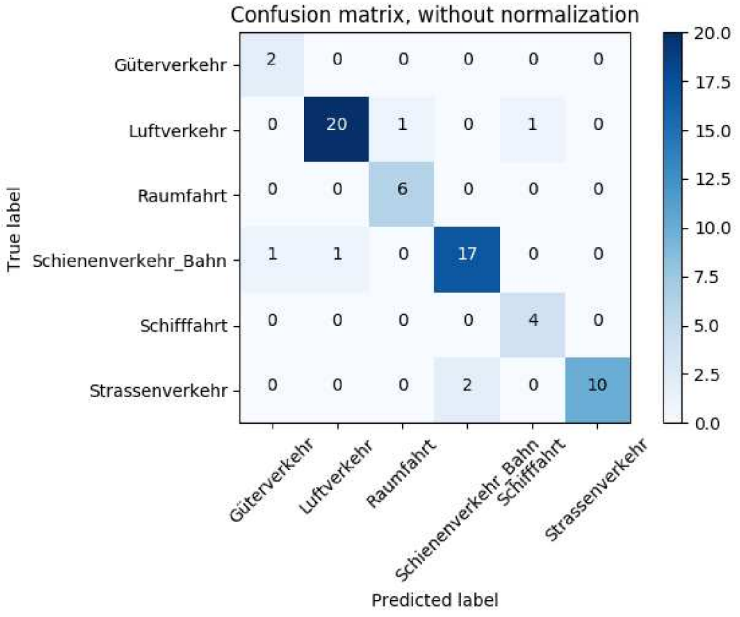


FIGURE 7.9: Confusion matrix for predictions of the ABCD-classifier for the domain *Verkehr (Traffic)* in absolute counts

7.5 General Remarks

We have chosen the task of document classification as it is one of the most prominent applications of language technology that is used in diverse areas. Additionally, document classification is an application of natural language processing that is applicable to many social science research scenarios. We focus especially on relatively small data sets (between 321 and 345 documents) with skewed distributions since this is a common pattern which is often not sufficiently tackled by standard supervised machine learning approaches.

The performance of the proposed approach is considerably successful in tackling the specific problems in such cases (i.e., accurate prediction also for small classes).

Therefore we conclude that the external generalization of the concept (or class) through the lexicon induction is helpful to solve the imbalanced data issue. The induced lexicons seem to aptly represent the concepts or classes, which leads to an encouraging performance. If we apply the lexical resources in an embedded modeling (ABCD), we surpass many of the other approaches, and sometimes even all of them.

To further make the approaches comparable, we chose a setting where we used the same training data set from a static stratified sampling for all contenders. This is important as we have used the possibility to derive the lexicon in a largely automated manner,

i.e., to use the same amount of annotated data as it was used to train a predictor for the other approaches. However, we want to state again that the predictor of the ABCD approach is heuristic and does not need any training at all, but certainly any serious application should at least provide annotated data for evaluation purposes.

The benchmark leads to several insights:

- In order to test for the "difficulty" of the static split, we compare it to the Baseline 80/20 and the Baseline CV5 classifier. The comparable performance of the cross-fold classifier points towards the comparable difficulty for the predictor for the task. In other words: we have not accidentally drawn an especially easy-to-solve sample in all three cases.
- When we compare the ABCD approach and the Baseline classifiers, we see that the baseline is outperformed for all cases.
- Since we also employed a classifier that relied on the same lexical resources but not in an embedded modeling, we are able estimate the benefit of the approach[12]. In two out of three cases, the embedding-based modeling (ABCD) is better. Additionally, it needs to be noted that this approach could be further improved by learning the predictor on the given labeled data. This has been intentionally left open to compare a basic and simplistic version of the predictor from ABCD which is still robustly successful. Lastly, it should be mentioned that nothing prohibits us from combining the purely lexical approach (look-up and counting) with the embedded modeling to get the best of both worlds (see also Chapter 8).
- To check the potential of improvement by setting the parameters for the heuristic predictor from ABCD appropriately, we applied a grid search over the two parameters (threshold t and n-best) and report the best performance in ABCD tuned. While there is an improvement for all three cases, it turns out that even the default values for the ABC approach lead to robust performance.
- With fastText NPV 80/20 we also controlled for a strong basic contestant. Without the access to the model of distributional semantics, i.e., the embedding, `fastText` was outperformed by ABCD.
- In terms of performance fastText 80/20— which also uses embeddings (including subword modeling)— was the best classifier. Although it must be noted that in the domain of *Education* this is not true; there, the ABCD approach was the best. Furthermore, `fastText` predicts only the label on the document-level (which is

[12] Of course, the lexicon-based classifier has a much simpler modeling that is based on the document as a unit and using the lexicons as bare feature generator via look-up.

the original task), in contrast to the prediction from ABCD on the sentence-level. Such a model is neither able to predict with the same performance on a smaller unit of data (sentences)[13] nor is it accessible for more fine-grained analysis (like the sentence-wise sub-concept detection from ABCD).

- Lastly, we also investigated with the fastText 20/80 classifier the reverse of the static split. Besides the domain of *Traffic*, we note that the decrease in training data leads to a remarkable drop in performance. This finding lets us also compare the performances of `fastText` and the ABCD approach if even less or no training data was be available.

Given the results from the benchmark, we conclude that also the rather simplistic projection from the sentence level to the document level is not problematic and suffices at least for this kind of classification problem[14]

7.6 Chapter Summary

In this chapter, we have presented a benchmark on three specific document classification tasks. These benchmarks were chosen to represent a challenging setting which incorporates a double-level skewness in the data. First, this scenario reflects cases which contain labeled data in the realm of a few hundred documents, therefore being of a much smaller size than other data sets. Second, the distribution between the classes is also harshly imbalanced, i.e., the task requires coping with very small classes (comprising of only a hand full of instances).

While the class imbalance problem is normally tackled with under-sampling (reducing examples from the most prevalent classes) and/or over-sampling (over-representing the small classes in the training set), this is not feasible in this case because of the distribution and size of the data set. This scenario with a small but skewed data set is therefore known to be demanding for standard machine learning approaches.

Indeed, as we observe, the baseline (SVM with TF-IDF) classifiers deliver acceptable performance with regard to accuracy. But they fail to represent at least one of the

[13] This holds true if we follow the premise that we could not just take all sentences of the documents and use the document-level label as label for the sentences. However, as this simplistic procedure may be usable for data sets which reflect the label throughout the document, it becomes highly problematic in the cases where the information for the label is strongly locally bounded or if we face a multi-label scenario.

[14] Although the approach offers a much more fine-grained analysis in the sense that we are able to infer *which* of the detectors—which are inspectable and aligned to sub-concepts that are found through the semantical clustering—was signaling at what strength for each sentence. Unfortunately, the lacking comparability with the other approaches renders this in-depth approach beyond the scope of the current chapter.

smaller classes for each case, which was made transparent by presenting not only micro measurements but also macro measurements. Furthermore, this fact was emphasized using confusion matrices to illustrate the difference between the baseline and the proposed approach (on the same data split).

In order to investigate the benefit of a more sophisticated modeling in contrast to a simplistic pure lexicon look-up we presented such a comparison against the ABCD approach.

Finally, we included `fastText` as a contestant, since it (optionally) leverages a model of distributional semantics, too, and is also geared towards solving problems stemming from data with difficult distributions.

Last but not least, we would like to emphasize again that the ABCD approach does not rely on labeled data and is therefore "unsupervised". For the sake of comparability, we attempted to set up the experiments to allow for a test against a (supervised) standard baseline.

In the same manner, the simplified modeling (that uses the same lexical resources like the ABCD approach but relies on counts from a look-up process) is a supervised classifier an thus requires annotated data. Although the `fastText` classifier had the best performance overall, we illustrated that in the case of less (training) data, the performance decreases as well. For the ABCD approaches we have used the data only for initial seed generation (to compare it against the baseline) and for tuning the two parameters that need to be set.

8

Experiments III: Framing Analysis Based on Concept Detectors

> "Facts are stubborn things, but statistics are pliable."
> — Mark Twain

```
In [93]: analogy(a="Wortwahl", b= "Nachricht",
x=None, y="Feuer", model_given=model, verbose=True)
'Wortwahl' is to 'Nachricht' as 'Flamme' is to 'Feuer'
Out[93]:
[('Flamme', 0.4252305328845978),
 ('Brand', 0.41508278250694275),
 ('rasch_löschen', 0.3844631314277649),
 ('Glut', 0.38384485244750977),
 ('lodern', 0.3811938762664795),
 ('Feuer_entfachen', 0.37259817123413086),
 ('Brand_geraten', 0.36972683668136597),
 ('Fahrlässigkeit', 0.36557522416114807),
 ('anzünden', 0.3634364604949951),
 ('Dachstock', 0.3621114492416382)]
```

In this chapter, we report on the application of different approaches to an intricate problem from the automated content analysis performed during a larger research project

(NCCR democracy[1]). While the overarching goal in the original content analysis was to study the interaction of four different variables (salience, issues, tonality, and frames), we will focus here on only one dimension, namely *frames*.

The main goal is to detect frames in media texts. More precisely, we focus on frames which refer to democratic legitimacy. For example, we would like to detect if the accountability of a regulatory body is the subject of a textual unit. The research interest here is if regulatory bodies and other new forms of governance which partly lack democratic legitimacy (because they are not elected and therefore this axis of democratic legitimacy is interrupted) are held accountable in the media, and if so, in which ways.

More technically, frames are rather latent features that refer to a specific aspect or perspective which is emphasized in the text. Frames often become manifest in the form of specific words or phrases (cf. Entman (1993): 52) but in other cases—and we will mainly deal with such a case in this chapter—the frame is more a schema for interpretation (cf. Goffman (1974))[2]. This schema is largely defined by the perspective that the author takes when writing the text. It allows the reader to quickly adopt this perspective to interpret the situation at hand, but, on the other hand, it also clearly influences the way in which a text is read and understood.

Since there is a wealth of studies suggesting a substantive influence on people's attitudes and opinions (see Chong and Druckman (2007a), and Chong and Druckman (2007b) for frames from multiple sources), we aim to be able to detect these frames in media texts. For the study which we carried out (with our collaborators in the project) several thousand documents have been annotated[3]. This data served firstly to develop a supervised model for the detection of such frames, and secondly to train a classifier which then predicted the kind of frame.

The annotators were asked to identify the core of the frame they annotated. This means that they marked the textual passage that was identifiable as "the core of the frame".

To give an example, see Figure 8.1 which depicts a screen-shot from the annotation tool *brat* (Stenetorp et al., 2012). This is a snippet from a text about the Kyoto Protocol, an international treaty that regulates the efforts, measurements, and goals which the signing countries want to achieve to protect against climate change[4]. Here we see that the whole sentence *Auf die Unterschrift der USA wartet die Welt noch heute. (Still today, the world is waiting for the US to sign the contract.)* was marked as being the

[1] http://www.nccr-democracy.uzh.ch/research/module1/IP6

[2] Further information on the framing approach that was taken for this study is given in Wüest et al. (2017). For a thorough discussion on framing approaches consider D'Angelo and Kuypers (2010) and Matthes (2014a); for a critical evaluation see Matthes (2014b).

[3] see Appendix C for the codebook and guidelines for the annotation

[4] see also `https://www.un.org/sustainabledevelopment/climate-negotiations-timeline/`

core of the frame. Additionally, we see that the annotator has determined that this is an efficiency frame (since it takes the United States so long to ratify the Kyoto Protocol[5]).

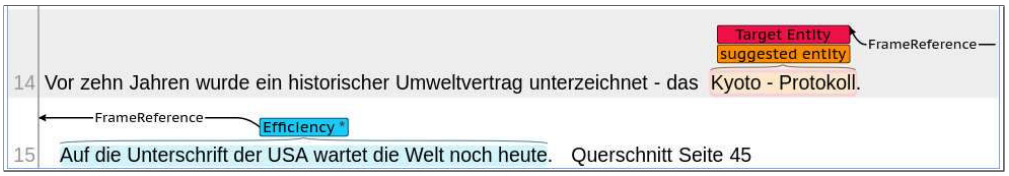


FIGURE 8.1: Screenshot from frame annotation with *brat*. The core of the frame is highlighted and linked to the entity of interest.

While this frame is rather implicit (one needs to know that waiting a long time is an indicator for inefficient processes), there are also much more explicit examples like

Wirtschaftsförderung betreibt zum Beispiel die Organisation Greater Zurich Area viel effizienter.

(The organization Greater Zurich Area is much more efficient in business development).

This sentence expresses explicitly that the evaluation of the business development carried out by the organization *Greater Zurich Area* is praised as being more efficient—which is then, rather clearly, an occurrence of an efficiency frame. Although it seems rather straightforward to detect such frames, we would like to point to the fact that many frames are rather latent or implicit features and their manifestation in text is often strongly locally bounded, sometimes even to a single word. The main point we want to make here is that we encounter a large variety when we inspect the annotated cores of the frames in the data set and that the automated detection and classification of such frames is not a trivial task. We will briefly address again the applied conceptualization of the frames and the operationalization for the annotation in the next section.

The remainder of this chapter is structured as follows: firstly, we briefly explain again the frames that we measured and provide a descriptive view on the data sets. Secondly, we evaluate the different approaches and their quantitative assessment whilst reflecting on the specific problems that have been tackled in different ways. Finally, we also provide a brief discussion where we focus on the proposed approach and argue why this is an apt solution to tackle the challenging task. Also, we will point to the limitations and shortcomings.

[5]Unfortunately, it turned out that the US did not ratify the contract at all. Worse, they even have opted out of the Paris Agreement, a follow-up treaty and international attempt to combat the harmful effects of climate change.

8.1 Frames of Legitimacy and the Core of the Frames

The frames that we were studying in the research project cover different aspects of democratic legitimacy. More precisely, the hypothesis underlying the framing study was that the public discourse about new forms of governance would contain specific perspectives considering their democratic legitimacy of (for a more detailed description see Amsler et al. (2016) and Wüest et al. (2017)).

We have already given a short explanation of the frames we aim to detect in Section 6.4.2.1 and how this conceptualization should be understood without including too many details. However, we briefly repeat here again the most important points:

- The sources for democratic legitimacy may be input-oriented, output-oriented and throughput-oriented (see Schmidt (2013)). The differentiation becomes clearer with the following short definitions:
  - **Input-oriented legitimacy frames** point to aspects that refer to who was involved in the decision-making process. There is a distinction on a more fine-grained level which distinguishes on aspects of representation (elected representatives), participation (involvement of citizens), deliberation (careful consideration and/or discussion), epistemicity (involvement of experts/scientists), and stakeholder inclusion.
  - **Throughput-oriented legitimacy frames** are present when the text focuses on aspects that describe attributes of the process itself. On the fine-grained level, we distinguish transparency (publicly available and disseminated information on the process), accountability (processes are controllable and correctable) and legality.
  - **Output-oriented legitimacy frames** represent aspects of the results of the political decision-making process. We differentiate here between efficiency frames (measures are efficient with regard to costs or time) and efficacy frames (measures lead to changes and solve the problems at stake).

Our annotators were trained to annotate all occurring frames in documents[6] and classify them with the 10-partite schema. The annotation was not restricted in the sense that

[6] After an initial training phase, the inter-annotator agreement was satisfying (micro-averaged F1-scores for fine-grained frame categories ranged between 0.66 (for 23 full documents during the annotation) and 0.71 (for 5 documents right after the initial training phase of the annotation).

multiple co-occurring and even overlapping frames were allowed. Hence the annotators were given the freedom to aptly annotate all perceived frames[7].

Additionally, we asked the annotators to also annotate the textual passage which led them to the annotation of the frame(s). While we have no access to this annotated passages during prediction, we used them to distill the typical vocabulary for the frames. To accomplish this, we again used the `SeedFinder` component—but now only referring to the marked textual passages as text for the frame classes (and not to the whole documents as in Chapter 7 for the document classification). Although this approach leads to a narrow focus on the used vocabulary, it allowed us to create a lexical resource that we generalized using the described algorithm in Chapter 4. For discussed examples of the derived resources, refer to the Section 6.4.2; a full overview on the resources is found in Appendix B.

8.1.1 The Task

The main task is best described as the automated detection of legitimacy frames in media texts and their classification. Since the detected frames should be linkable to specific named entities in the text[8], we adapted the task so that the prediction would be carried out on the paragraph-level.

This also met the empirical outcome of the annotation process which revealed that the textual passage marked as "core of the frames" was often shorter than a sentence but sometimes also spanned over several sentences. However, the annotations of the textual passages did not cross the borders of paragraphs. This is plausible in the sense that paragraphs contain semantic units which convey statements.

While we had access to the textual passages being identified as the "core of the frames" for training, we would not have any knowledge about the passages during prediction time. This means, while we knew which part of the paragraph was carrying the main trigger for the frame annotation, we needed to create a system that does not require this information for prediction.

[7] A more encompassing operationalization of the (manual) content analysis is referenced in Appendix C.

[8] Note that we collected (most of) the texts with the query for specific named entities of interests (i.e., organizations, institutions, companies, treaties) in the first place. Our data acquisition resulted in collections of documents. After preliminary tests, it turned out that we would need to make sure that there is a linkage between the named entities in the query and the frames we detected. Otherwise, we would erroneously attribute frames to the entities of the query (in the case that other entities are the target of the framing). In order to do so, we narrowed the unit of prediction to the paragraph-level, so that we predicted the frames per paragraph. This enabled the researchers to check if their named entity of interest was occurring in the same paragraph. Or to view it from a different perspective: they had the possibility to filter for the frames occurring only in paragraphs where their named entity of interest was present.

However, in order to make use of the passages, we used them to distill the seed for the lexical induction we described in Section 6.4.2. Furthermore, the task was finally restricted to predict only one frame per paragraph[9]. With the benefit of hindsight, one would have avoided this restriction, as this was not in line with the annotation guidelines which stated that any number of frames could be annotated in the same paragraph[10].

Due to the variance in the length of the paragraphs in the different media outlets which were part of the analysis (long reports in weekly magazines tend to have much longer paragraphs than short online articles), we also encountered the situation of having multiple instances of the same frame within longer paragraphs. There are at least two causes for this: firstly, the paragraph may contain the same frame which is referring two separate cases, i.e. two separate named entities. While one could also take the position to unite these occurrences of that frame over the paragraph, this is an insufficient solution given that we would want to align the frame with a named entity. Secondly, there may be two instances of the same frame that refer twice to the same entity. While this is a less problematic case for the "union" of the two frame instances, the frames are often used in a contrasting juxtaposition, for example, if the consequence of a regulation is evaluated in one case as effective while being ineffective for other cases. This is not a problem *per se*, since both instances of the frame are of the same category. But in order to prevent flawed data sets (with multiple identical data points—in this case only the text of the paragraph—that have the same target label), we needed to filter out those instances of frames.

To address the issues outlined above, we will carry out an additional separate evaluation for the differentiation of the text passages alone (i.e., the spans of text marked as the "core of the frames" which we also call frameslices). This should reveal more insights about the discriminatory power of the system with regard to the distinction between the actual "cores of the frames".

If not stated otherwise, the provided quantitative evaluations in this chapter are based on the paragraph-level and refer to the single-label task, i.e., assigning only one frame (the most prominent) in case of occurrence.

Since the defined frames have conceptual overlaps, we also report the results of a coarse-grained, tri-partite conceptualization, where the frames are only predicted as being

[9]While this restriction was meant to simplify the aggregation processes of the analyses, it caused a divergence between the way the data was annotated and the desired prediction. Unfortunately, this complicated the evaluation in various ways in the sense that we had to enforce a decision which frame was the "chosen" label if multiple frames were present. Furthermore, if the same frame was present multiple times, we had to make sure to retract this sample from the test instances in order to prevent information leakage. In this way, we had to retract the paragraphs with the clearest triggers (being multiple instances of the same frame).

[10]We even allowed for multiple frame annotations given the same piece of text. This covers the situations where two frames appear interweaved.

input, output, or throughput frames. This case will be referenced as "coarse-grained" (in contrast to "fine-grained" which refers to the 10-partite categorization).

We will also report on the binary case, where we only decide *if* a frame is present in a paragraph (no matter which fine-grained or coarse-grained category) The rationale behind this is that on the one hand, there is also some overlap between the coarse-grained categories, and, on the other hand, the skew in the data distribution (see next section) lead us to tackle this challenge with a two-level approach, where we first decide if there is a frame and in a second stage differentiate the categories.

8.1.2 The Data Sets

In the NCCR democracy project, we conducted a project-overarching study and annotated frames in 1.951 documents in German. This resulted in 26.421 paragraphs where we annotated occurring frames or (implicitly) labeling them as containing no frames[11].

Since 1.929 annotated paragraphs contained more than one frame per paragraph, the total amount of annotated frames is 29.195 (this also contains the implicit annotation for NOFRAME). From this initial data set, we performed a stratified split with regard to the frame categories and created a training set of 26.275 units and a test set of 2.920 units (i.e., a 90/10 split). The test set was never inspected nor used before the final evaluation.

Table 8.1 displays the distribution over the different classes for the training set[12]. Note that we also include labels which point to the cases in which the annotators had not decided on the fine-grained label for the frame. In those cases decision was to fall back on the coarse-grained level (using *Input_Legitimacy*, *Output_Legitimacy*, or *Throughput_Legitimacy*), or even on the binary level (using *Democratic_Legitimacy* as a generic label)[13]. We notice that the residual class, i.e., the case where there is no legitimacy frame present in the paragraph, is by far the most prominent class.

More precisely, we find that over 72% of all annotations concern paragraphs not containing any frame. This means that we have a heavily skewed distribution that would lead to a ratio of 151.6 between the largest class (*NOFRAME*) and the smallest (*Accountability*) if we tried to integrate the detection and classification of frames into one step. Certainly, we have to provide counter-measurements to tackle this skewness.

[11]We thank the involved annotators Michelle Ammann, Anna-Lina Müller, and Anna Sigrist for their valuable contribution to this research project

[12]The test set displays exactly the same distribution—trivially because of the stratified split.

[13]While we certainly avoided including any artificial additional classes—as we would subsume the fine-grained labels with the coarse-grained labels according to the schema presented in 6.4.2.1—we used those examples which were only labeled on the coarse-grained or binary level as supplementary data points for the respective classification.

Frames annotated in German corpus			
Label	**n**	**% of Annotations**	**% of Frames only**
NOFRAME	18946	72.11	-
Efficacy	2552	9.71	36.00
Representation	1232	4.69	17.38
Efficiency	1082	4.12	15.26
Stakeholder	746	2.84	10.52
Participation	357	1.36	5.04
Epistemic	271	1.03	3.82
Transparency	270	1.03	3.81
Legality	253	0.96	3.57
Deliberation	201	0.76	2.84
Accountability	125	0.48	1.76
Democratic_Legitimacy	112	0.43	-
Output_Legitimacy	110	0.42	-
Throughput_Legitimacy	14	0.05	-
Input_Legitimacy	4	0.02	-

TABLE 8.1: Distribution of the frame annotations over the training set from the German corpus

When we analyze the distribution further, we also see that we have a large skew *within* the frame classes. For example, the ratio between the largest frame class (*Efficacy*) and the smallest (*Accountability*) is still 20.4. This calls for another specific solution to counteract the problem of the skewed distribution.

We call this situation where we encounter this imbalance in the data *double data skew*: the first skew is one on the binary level where we decide if there is a frame present at all[14]. On the one hand, this skew may be tackled with downsampling techniques. On the other hand, since the majority class is the residual class, the question remains open if there are distinctive features of those data points that make them a member of that residual class besides the absence of frames. If there are such features, it would be important to take them into account while downsampling.

The additional second level of data distribution skewness—the distribution over the frames only—poses an extra challenge, especially if the architecture for the predictor is flat, i.e., if we predict all fine-grained frames directly (and do not integrate a prior first prediction on the coarse-grained level).

While we will present the quantitative evaluation for the data set we described above, we will will additionally also report on the results on a second, smaller evaluation data

[14]Note that such a binary conceptualization is only feasible if the classes share a generalizable common core or if they can be modeled all together in an additive way without contradiction.

set which was based on more constrained sampling criteria than the original data set[15]. This evaluation set was not used during development of the framing detection and served only to evaluate a specific subset of texts. Although the data set is much smaller, we consider it worthwhile to report the results on this set as well.

This data set contained annotations for 503 paragraphs for German texts. Table 8.2 shows the distribution over the frame labels. Note that the generic categories were not used for this annotation. Interestingly, while we had a very similar distribution over the categories—although less paragraphs with no frames— in comparison with the original set (see Table 8.1), we had no occurrence of the *Deliberation* frame in the data set.

Frames annotated in German Evaluation Data Set			
Label	**n**	**% of Annotations**	**% of Frames only**
NOFRAME	287	57.06	-
Efficacy	117	23.26	54.17
Representation	30	5.96	13.89
Efficiency	28	5.57	12.96
Stakeholder	15	2.98	6.94
Participation	9	1.79	4.17
Legality	9	1.79	4.17
Transparency	4	0.80	1.85
Epistemic	3	0.60	1.39
Accountability	1	0.20	0.46
Deliberation	0	0.00	0.00

TABLE 8.2: Distribution of the frame annotations over the smaller evaluation data set

To summarize again the insights gained from the description of the data sets and their consequences:

- We annotated frames of legitimacy in media texts
- The frames are annotated in a 10-partite schema, i.e., we have ten different frames that we annotated. This does not include the residual class (NOFRAME) which represents the cases when no frame is present.

[15]The data, i.e., the documents for this data set were chosen based on the occurrence of specific entities of interest in the texts. For the original data set, we sampled from a wide range of different sub-projects from the researchers we collaborated with in this project. The goal was to create a resource to detect frames for a broad variety of entities of interest. However, since some of the sub-projects were given special attention for specific case studies, the need arose to evaluate on text from the specific entities. This means that we sampled only from the respective sub-corpora. Since this evaluation was done in a later stage of the project, also another person was hired to carry out the annotation. We thank Rolf Badat for his valuable contribution to this project.

- The unit of analysis is the paragraph and not the document. Although not completely in line with the assumptions we have made while annotating, the task was defined to predict the most prevalent frame (or the absence of frames) per paragraph.
- The residual class (NOFRAME) is by far the most prominent
- Also on the level of frames, there is a large skew in the distribution
- Since we have a double data skew, we firstly modeled the task as a binary one, i.e., we created a classifier to decided if there are *any* frames in a paragraph—or no frames.
- In addition to the fine-grained prediction we also produced a coarse-grained prediction which groups the frames classes into input, output and throughput. This enables us to estimate the performance on this aggregation level directly.
- Since the occurrence of multiple frames in the same paragraph is possible, we have to filter those points from the data sets in order to prevent an information leakage
- To analyze the differentiability of the marked textual passages, we evaluate this scenario separately

8.2 Quantitative Evaluation

In this section we will report on the quantitative evaluation of the approaches. We therefore compare a baseline approach—which partially failed due to several task-inherent problems—to the one which was applied in the end to generate the analysis for the project. Additionally, we will also provide measurements of an approach which is based on ABCD alone for the same evaluation data sets. This means, we applied the proposed approach of this work ex-post to the same data we have used during the project. While the approach that we applied in the project has a larger overlap with the ABCD approach, it relies heavily on supervision and is strongly tuned in different ways which makes it less generic than ABCD.

8.2.1 Applied Approaches

In this subsection we briefly describe the applied approaches. While we will not investigate further the potential improvement of the baseline—which we will mainly use to make evident that standard models run into several problems for this task—we quickly

introduce the approach that was applied in the research project from NCCR democracy (we call this the "SIFT approach", which is the name of a paper on the project's results).

Additionally, we also apply the proposed method from this work, the ABCD approach. We applied this approach in its pure form ex-post to the same data—especially also to find out how it will perform against a system which was heavily tuned by supervised machine learning techniques.

However, since the ABCD approach relies on a subset of the features of the SIFT approach but is parametrized with only two parameters, this is also an interesting comparison for the predecessor (SIFT) and its newer form (ABCD). While SIFT is fully supervised, ABCD (in the presented form) does not incorporate any learned decision function given its features but predicting according to its basic heuristics.

8.2.1.1 Baseline Approach

We created a baseline approach based on the Stanford Classifier[16] which incorporated standard text classification techniques and preprocessing steps[17] for comparison reasons.

Since it was evident from the first experiments that the skewed data distributions would be problematic for the baseline system—in the sense that we get a low recall for the smaller class of frames—we will only report on its results from the experiments on the original data set and not on the secondary evaluation data set where we set the focus on the comparison between the SIFT and the ABCD approach.

8.2.1.2 SIFT Approach

In the following, we describe the approach we referred to in Wüest et al. (2017).

On the one hand, we calculate lexical features with the lexicons described in 6.21. This results in a count of occurrences of words from the respective lexicon that represents the concept of a fine-grained frame category[18]. Additionally, we also calculate the ratio of tokens that belong to a lexicon to control for paragraph length. This results in 20 features (10 count-based, 10 ratios).

On the other hand, we applied a similar technique as described in Chapter 7 for the text classification with the ABCD approach, i.e., we embedded a filtered part of the lemmatized sentences into the semantic space and compared them (sentence-wise) to

[16] See `https://nlp.stanford.edu/software/classifier.shtml`

[17] We applied only lowercasing as preprocessing and added bigrams and 3-grams

[18] If summed together, we control for 3.395 words that occur in the ten lexicons.

the concept detectors (i.e., the centroids of the clustered re-embedded lexicons). To be clearer and also point to subtle differences (starting with point 4.), we repeat here again the most important steps

1. In a first step we split the data point (which is the text contained in a paragraph) into single sentences.
2. These sentences are then preprocessed and filtered for the lemmata of a subset of part of speech, i.e., nouns and adjectives.
3. This filtered and lemmatized version is then compared to the concept detectors, i.e., the derived clusters centroids of the embedded lexicons—which represent them in the semantic space
4. We create several scorings for the whole paragraph, mainly relying on the similarity scores of the sentences (7 features)
5. We sum up the similarity scores for the fine-grained frame categories (10 features)
6. We sum up the similarity scores with respect to the coarse-grained frame-categories (3 features)
7. We also add the overall sum of similarities to the different frame concepts as a feature (1 feature)

Overall, we create 41 features in this way that we use to learn the models. Through extensive testing of different settings, we decided to apply a soft ensemble of a Random Forest, a Logistic Regression and a NaïveBayes classifier. This scenario may look a bit complicated at first sight.

The goal is to combine at least three different purposes: firstly, the logistic regression learner performed acceptably well and—since the features are all written in a positive way—its regression coefficients were useful to further inspect the contribution of the different features in a straightforward way. Second, the Random Forest was included to be attentive to cases in which special combinations of values should lead to a specific prediction which were not directly inferable for the global model of weight attribution from the logistic regression. Lastly, the NaïveBayes classifier was originally included to exert influence indirectly on the ensemble's decision by including slightly shifted priors for the small classes. Finally, this tweak was realized with other techniques. But nevertheless, the results of the ensemble tended to be more stable with the inclusion of the NaïveBayes predictor.

To counteract the problems even on the dichotomous binary level (low recall of the frames due to the overly present residual class), we additionally applied firstly a downsampling step in an informed way. [19] Therefore, we firstly created a score for each data point which subsumed the features that were created to have a high correlation with the occurrence of the frames. Secondly, we used this score as a threshold to filter out data points from the residual class, so that mostly clear-cut examples remain in the train set. The actual threshold—which naturally influenced the number of data points of the residual class—was empirically determined through extensive grid search in cross-validation results in the train set. Consequently, this choice may also be seen as 'data fit' but it is at least tentatively made more robust through the cross-validation. The rationale behind this was not to induce any information leakage from the test set to the parameter selection. In this case, the adoption of the best downsampling threshold to reach a better result regarding the test set would have created such a leakage.

8.2.1.3 ABCD Approach

While we used the same lexical resources as for the SIFT approach, the prediction schema for the target label was slightly different. Since the ABCD approach is based on positive signals (there are no lexical resources for the NOFRAME category), we needed to define a threshold beyond which we would label the given data point as not containing any frame. In other words, if the signals we measure (through the summed similarity scores with regard to the concept detectors) are not high enough, we simply predict that there is none of the frames in the paragraph. This rule can be boiled down to the premise that there is nothing if we have no signal for it.

8.2.2 Benchmark

In this section we finally compare the quantitative performance of the applied approaches. We benchmark them on three different data sets.

Firstly, we apply them to the test set of our main data set, where we have developed and trained the classifier (on the training set). Secondly, we apply it to the second evaluation set we created. Lastly, we will also report the performance on the prediction of the annotated slices of text which were marked as "the core of the frame".

Although the results on the last data set are not to be transferred to the real application (since, normally, these text passages are not annotated in the first place, and, even more

[19] We call this an *informed way*, since we intentionally decide which data points—or better which kind of data points—we exclude from the training set while downsampling the abundant residual class.

important: they are not available for the prediction anyway), we use them as a proxy to estimate the quality of prediction. If we had have access to the information of the exact textual passage, the prediction would be performed on this unit of text only—since this slice of the text contains the necessary information. The gap in performance between this prediction and the full illustrates how strongly locally bounded information gets partially lost in the modeling of single-labeled paragraphs.

In Table 8.3 we show the performance on the binary task, i.e., to predict if *any* of the given frames occurs (class FRAME) or if the paragraph does not containing any frames (class NOFRAME). The test set contains 2920 instances and has the same distribution as reported in Table 8.1, i.e., we have over 72% paragraphs which do not contain any frame. We observe that the accuracy (0.77) of the Baseline model is not bad. But when we inspect the performance measurement more closely, we see that although the precision for the FRAME class is acceptable (0.64), the recall is rather low (0.39). This is a heavy drawback since the detection of the frames is the main purpose of the model and therefore it is not an option to trade-off the good overall accuracy for the low recall of the frames from which we would miss more than 60%.

When we look at the SIFT performance, we see that the recall for the FRAME class is notably better (0.54), although still leaving room for improvement. Additionally, we see a slight decrease in precision for the FRAME class, while we reach almost the same accuracy (0.76 vs. 0.77) compared to the baseline. Since we also almost maintain the same F1 score for the NOFRAME class (0.83 vs. 0.85 in the baseline)—loosing a bit on recall (0.84 vs. 0.91) but gaining on precision (0.82 vs. 0.79)— we also see a small increase in the macro average of the F1 score (0.69 over 0.67 from the baseline).

To boost the recall even further (on the cost of precision) we also trained a model so that the recall of the FRAME and the NOFRAME class approximately even out. We labeled this approach $\text{SIFT}_{\text{recall\_boost}}$. As we observe, we reach a recall on the level of 0.70. While we accomplish this high recall for the FRAME class, the recall from the NOFRAME class decreases in the same time from 0.84 to 0.71 (and also the precision for the FRAME class decreases from 0.57 to 0.48). However, we are able to boost the recall of the FRAME category while maintaining an overall accuracy (which is sensitive to the performance of the over-represented residual class) of 0.70.

In the next step, we compare the performance from two implementations of the ABCD approach to the others. Note that the prediction of the ABCD approach in this form is not tuned or learned using the labeled data; it simply relies on the heuristic prediction

Approach	Class	Prec.	Rec.	F1	Number of Instances
Baseline	FRAME	0.64	0.39	0.48	814
	NOFRAME	0.79	0.91	0.85	2106
	Micro avg.	0.77	0.77	0.77	2920
	Macro avg.	0.72	0.65	0.67	2920
SIFT	FRAME	0.57	0.54	0.55	814
	NOFRAME	0.82	0.84	0.83	2106
	Micro avg.	0.76	0.76	0.76	2920
	Macro avg.	0.69	0.69	0.69	2920
$\text{SIFT}_{\text{recall\_boost}}$	FRAME	0.48	0.70	0.57	814
	NOFRAME	0.86	0.71	0.77	2106
	Micro avg.	0.70	0.70	0.70	2920
	Macro avg.	0.67	0.70	0.67	2920
ABCD	FRAME	0.43	0.72	0.54	814
	NOFRAME	0.85	0.63	0.73	2106
	Micro avg.	0.66	0.66	0.66	2920
	Macro avg.	0.64	0.67	0.63	2920
$\text{ABCD}_{\text{even\_recall}}$	FRAME	0.43	0.66	0.52	814
	NOFRAME	0.84	0.67	0.74	2106
	Micro avg.	0.66	0.66	0.66	2920
	Macro avg.	0.63	0.66	0.63	2920

TABLE 8.3: Evaluation of the Binary Prediction Task on the Original Test Set

based on the two parameters (threshold t and number of n-best[20]). We see that for the normal ABCD approach, we reach a recall or 0.72 for the FRAME class, although the precision is as low as 0.43. Furthermore, the performance for the FRAME class is also partly reached through the partial sacrifice of the performance in terms of F1 scores on the residual class NOFRAME which is on 0.73.

We include the report on a slightly modified approach we name $\text{ABCD}_{\text{even\_recall}}$. With this additional setting, we illustrate how we almost directly trade off recall (from the FRAME class: down from 0.72 to 0.66) for the recall of the other (recall of the NOFRAME class increases from 0.63 to 0.67)[21].

Overall, we may summarize that we are able to tackle the low recall problem for the FRAME class from the baseline approach—although at the cost of the performance on the residual class. However, with the SIFT approach, we increase not only the recall for FRAME class, but we also increase the macro average scores while we almost maintain the overall accuracy. While the SIFT approach performs best, it is also interesting to see how well the ABCD approach performs, given that it does not leverage the labeled data to its full potential. Also, the $\text{ABCD}_{\text{even\_recall}}$ performance illustrates how the approach allows for (transparent) flexibility by changing only the two global parameters.

For the next experiment, we turn now to the prediction on the coarse-grained level. This leads then to a three-partite classification which differentiates between the (grouped) frames. Note that for this task, we only take the data points into consideration which *are* frames. This means, we only measure how good the classification is for paragraphs containing frames[22].

In Table 8.4 we observe that the SIFT approach as well as the ABCD approach outperform the baseline by a large margin. This holds true for the scores in precision, recall, and F1, as well as for the overall accuracy. It is again one of the prevailing undesirable

[20]These two parameters are set at 0.62 for the threshold, and n-best to 10. We picked those values according to the performance levels (accuracy and F1 score) we wanted to compare. Accordingly, on could also argue that in a strict sense, these values are then "tuned" using the labeled data as well. However, to really benefit from the labeled data, one would rather introduce a weight matrix for the single concept detectors and not optimize only the two global parameters. Of course, the interplay must then be handled as well.

The parameters for the $\text{ABCD}_{\text{even\_recall}}$ are 0.63 for the threshold and 10 for n-best. This also shows the possibility to gradually tune the performance of the ABCD approach: by slightly raising the threshold, we get, as a consequence, less false positives for the FRAME class, and, accordingly more true positives for NOFRAME. But we also get less true positives for the FRAME class which decreases the recall. This is hence a classical precision-recall trade-off in such a binary case where we just predict the residual class in the case of the absence of a signal directly reflected in the threshold.

[21]Of course, the difference in the numbers is different due to the skewed distribution of the classes.

[22]This selection of measurement seemed natural due to the fact that we implemented a stacked classification for the research project, i.e., first we decided if the paragraph contains a frame and in a second step classify the frame, *only if* the first stage predicted a frame. To compensate for this methodological drawback which disallows comparisons with systems that deliver the full prediction in one step (including the residual class NOFRAME), we present according measurements later in Table 8.9

Approach	Class	Prec.	Rec.	F1	Number of Instances
Baseline	Input_Legitimacy	0.42	0.42	0.42	312
	Output_Legitimacy	0.57	0.59	0.58	416
	Throughput_Legitimacy	0.15	0.12	0.13	74
	Micro avg.	0.48	0.48	0.48	802
	Macro avg.	0.38	0.38	0.38	802
SIFT	Input_Legitimacy	0.58	0.56	0.57	312
	Output_Legitimacy	0.70	0.71	0.71	416
	Throughput_Legitimacy	0.39	0.42	0.41	74
	Micro avg.	0.63	0.63	0.63	802
	Macro avg.	0.56	0.56	0.56	802
ABCD	Input_Legitimacy	0.63	0.46	0.53	312
	Output_Legitimacy	0.66	0.79	0.72	416
	Throughput_Legitimacy	0.37	0.39	0.38	74
	Micro avg.	0.62	0.62	0.62	802
	Macro avg.	0.55	0.55	0.54	802

TABLE 8.4: Evaluation of the Prediction Task for Coarse Categories on the Original Test Set

characteristics of the baseline (which is standard text classifier) that it performs poorly on the smaller classes (here *Throughput_Legitimacy*). While this outcome is often bound to the skew in distribution over the classes, in this case we also see a large improvement on the largest class (*Output_Legitimacy*) in the performance from the other classifiers.

Overall, this measurement exemplifies where the SIFT approach (and hence the ABCD approach) unfold their biggest potential: to discriminate between unevenly distributed classes. The improvement against the baseline is remarkable, be it in terms of accuracy (0.48 to 0.63 and 0.62 respectively) or macro averaged F1 scores (0.38 to 0.56 and 0.54 respectively). Nevertheless, the gap in terms of performance between the smallest and the biggest class still persists although on a more acceptable level.

In Table 8.5 we turn to the case where we predict the fine-grained categories of the frames. Hence, this is a 10-partite classification. We report in this case again the performance of a baseline, the SIFT approach and the ex-post measured performance of the ABCD approach, but rely on the averages of micro and macro levels for the sake of clarity.

We show that the baseline is clearly outperformed by the two given approaches. Especially the macro averaged scores illustrate the improvement. This points to the more evenly distributed benefit across all classes (the improvement on the micro averaged level is also consistent but remains to be a bit lower throughout)—or, to see it from another perspective: we improve on the poor performance for the small classes which causes the baseline to produce the low macro averaged scores.

	P_{micro}	P_{macro}	R_{micro}	R_{macro}	$F1_{micro}$	$F1_{macro}$	Acc
Baseline	0.30	0.18	0.30	0.17	0.30	0.17	0.30
SIFT	0.38	0.30	0.38	0.33	0.38	0.31	0.38
ABCD	0.40	0.31	0.40	0.26	0.40	0.27	0.40

TABLE 8.5: Evaluation of the Prediction Task for Fine Categories on the Original Test Set

We further note that for the given problem of fine-grained classification of paragraphs, the ABCD approach has an even higher accuracy than the SIFT approach. But this is partly rooted in the heavy tuning of the latter to produce a high recall scores for all the classes. In this point, the SIFT approach clearly delivers the best result (0.33 vs. 0.26 from the ABCD approach) and hence also has the best macro averaged F1 score (0.31 vs. 0.27 from the ABCD approach).

When we sum up the insights gained from the different measurements, we consider the detection of frames as a challenging task and even the sub-task of differentiating between the frames is demanding. We showed that a standard text classification baseline was outperformed in the binary detection task where it suffered from low recall considering the class of interest (FRAME). Although the performance in the prediction for the residual class (NOFRAME)—for which there is no conceptualization besides the absence of the concepts of interest but which needs to be modeled or "annotated"[23] in any case for a discriminative model—slightly decreased, this improvement allowed to carry out the automated analysis for the research project with satisfactory quality[24]. We observe that the quality of the classification itself (i.e., only the differentiation *between* different frames) naturally decreases the finer the categories are (from an accuracy of 0.63 for the coarse-grained case to 0.40 for the fine-grained case), but we still improve on almost all levels against the baseline.

[23]This was implicitly done in our case through not annotating any frame in the paragraph.

[24]Since the results of the binary prediction were delivered with probabilities for both classes, the other researchers who built their analyses on this data were enabled to decide ex-post which way fitted their needs best. For example, if they opted to be more cautious, they would only accept predicted frames with a probability above a given threshold that is higher than 0.5. Or they were able to couple this information with the restriction that a frame should be present in multiple paragraphs which was true for the majority of all cases.

In the following part of the benchmark, these results from the evaluation on the test set from the original data set will be now be complemented with the results on the second evaluation set. Since it has been shown that the baseline is outperformed in almost all cases, we will concentrate on reporting the results from SIFT approach and the ABCD approach on the second evaluation set. Furthermore, we compare those to the results on the first test set.

As we have stated earlier, the original data set was created by annotating a sample of articles which comprised of many different entities and, hence, also differed strongly in thematical focus and its textual manifestation. Since we developed, learned and evaluated the framing detection on this data, the quantitative evaluation contained still an amount of uncertainty how well it would work in specific cases (i.e., with specific data sets which are themselves compiled through the retrieval given some queries relating to specific named entities). To answer this question, a second evaluation set was assembled which represented the entities of pivotal interest for the other research groups. Although considerably smaller (we have 503 annotated paragraphs for German articles), we nevertheless consider it insightful to report on those results. This especially allows us to compare and illustrate the performance on different data sets.

We will stick to the top-down evaluation scheme and present first the performance on the binary task, i.e., the detection of frames. Second, we will report on the coarse-grained and the fine-grained classification. This part is intentionally held analogous to allow for comparison. Additionally, we will also include a measurement of the approaches which includes the full prediction, i.e., we do evaluate the performance for coarse-grained and fine-grained prediction but also include the residual class NOFRAME. This allows us to estimate the quality of the approach given the original task: to label paragraphs with the most prevalent frame (or label them as not containing any frame).

When we look at Table 8.6 we observe that the results are noticeably better than on the other test set. Here, we are able to reach a remarkably high recall for the FRAME class of 0.90 while maintaining an acceptable recall of 0.75 (SIFT) and 0.70 (ABCD), thus leading to high F1 scores. Note that this data set contains more paragraphs with frames (43% vs. 28% in the test set from the original data set). While less skewed distributions often offer a less challenging scenario to learn a discriminative classifier, this also hints at the fact that the mere selection of the content (i.e. the articles), may also introduce a—or better: determine the—specific level of difficulty to keep the classes apart.

This trend of increased performance is also consistent with the next measurement where we look again at the differentiation of the frames (n=216) into the coarse-grained categories *Input_Legitimacy*, *Output_Legitimacy*, and *Throughput_Legitimacy*. In Table 8.7 we observe that the overall accuracy is 0.68, which is slightly higher than in the same

Approach	Class	Prec.	Rec.	F1	Number of Instances
SIFT	FRAME	0.73	0.90	0.81	216
	NOFRAME	0.91	0.75	0.82	287
	Micro avg.	0.82	0.82	0.82	503
	Macro avg.	0.82	0.83	0.81	503
ABCD	FRAME	0.70	0.90	0.78	216
	NOFRAME	0.90	0.70	0.79	287
	Micro avg.	0.79	0.79	0.79	503
	Macro avg.	0.80	0.80	0.79	503

TABLE 8.6: Evaluation of the Binary Prediction Task on the Second Evaluation Set

case, the coarse-grained classification, in the original test set (0.63 (SIFT) and 0.62 (ABCD); see Table 8.4).

Also, we measure a recall of 0.6 or above for all classes in both classifiers. This leads in turn to higher macro averaged recall scores (0.56 vs. 0.73 for SIFT; 0.55 vs. 0.67 for ABCD) and is therefore linked with the higher macro averaged F1 scores (0.56 vs. 0.60 for SIFT; 0.54 vs. 0.58 for ABCD). Whilst the recall scores for the coarse-grained classes do not differ much anymore, this is partly bought at the cost of lower precision which is in this measurement the most remarkable drawback: especially the precision for *Throughput_Legitimacy* is low: 0.24 (SIFT) and 0.26 (ABCD), compared to the values from the original test set: 0.39 (SIFT) and 0.37 (ABCD). However, the overall result is still better for this second evaluation set.

When we turn to the evaluation of the fine-grained classification, the difference in the measurements between the both test sets is even larger. In Table 8.8 we note that the accuracy is at 0.54 (SIFT and ABCD) and therefore much higher than for the original test set (0.38 for SIFT and 0.40 for ABCD). Also, we would point to the higher values for macro averaged precision (0.44 vs. 0.30 (SIFT) and 0.37 vs. 0.31 (ABCD)), macro averaged recall (0.42 vs. 0.33 (SIFT) and 0.40 vs. 0.26 (ABCD)), and macro averaged F1 scores (0.36 vs. 0.31 (SIFT) and 0.34 vs. 0.27 (ABCD)) as well.

In sum, the evaluation on this second evaluation data set yielded generally higher scores[25] and this finding was consistent for all levels of prediction (binary, coarse-grained, and

[25] As noted in the description of the data sets: the secondary evaluation set was annotated by another person than the original data set where three annotators contributed to the project. This may also have influenced the performance measurement, in addition to the selection of the content by restrictions on certain entities.

Approach	Class	Prec.	Rec.	F1	Number of Instances
SIFT	Input_Legitimacy	0.67	0.67	0.67	57
	Output_Legitimacy	0.89	0.66	0.76	145
	Throughput_Legitimacy	0.24	0.86	0.37	14
	Micro avg.	0.68	0.68	0.68	216
	Macro avg.	0.60	0.73	0.60	216
ABCD	Input_Legitimacy	0.62	0.60	0.61	57
	Output_Legitimacy	0.83	0.70	0.76	145
	Throughput_Legitimacy	0.26	0.71	0.38	14
	Micro avg.	0.68	0.68	0.68	216
	Macro avg.	0.57	0.67	0.58	216

TABLE 8.7: Evaluation of the Prediction Task for Coarse Categories on the Second Evaluation Set

	P_{micro}	P_{macro}	R_{micro}	R_{macro}	$F1_{micro}$	$F1_{macro}$	Acc
SIFT	0.54	0.44	0.54	0.42	0.54	0.36	0.54
ABCD	0.54	0.37	0.54	0.40	0.54	0.34	0.54

TABLE 8.8: Evaluation of the Prediction Task for Fine Categories on the Second Evaluation Set

fine-grained).

In addition to the evaluations that we have presented so far for the first and the second evaluation set, we now also shed light on a measurement which is geared to estimate the performance on the full prediction task, i.e., not only focussing on the binary detection task or the differentiation task. Thus, we include the NOFRAME category (or the data points of that category) into the evaluation scheme. We hence take all the 503 paragraphs into account, which include the 287 paragraphs containing no frames.

Focusing first on the coarse-grained classification[26], we may compare the results of the first row from Table 8.9 with the ones from Table 8.7. For the SIFT approach, the accuracy even increases (0.72 vs. 0.68) but the macro averaged scores for precision, recall, and F1 all decrease. For the ABCD approach, the accuracy remains at 0.68, but also her we see a slight decrease of 0.03 in macro averaged precision, recall, and F1. The decrease in the macro averaged measured was to be expected as a step from a three-partite to a four-partite classification normally leads to lower scores if averaged over all

[26]To be precise, this would mean in this case to predict *Input_Legitimacy*, *Output_Legitimacy*, *Throughput_Legitimacy*, or *NOFRAME*.

classes with same weights. On the other hand, the decrease is rather low (besides the drop for macro averaged recall for the SIFT approach from 0.73 to 0.63) and the overall accuracy of the (full) prediction task is at 0.72 (SIFT) and 0.67 (ABCD). In other words, at least two-thirds of all paragraphs would be labeled correctly.

When we look at the results from the fine-grained classification (including the prediction of the residual class) in the second row in Table 8.9 and compare it with the results from Table 8.8, we recognize a similar picture. Again, we have an increase on accuracy (0.54 to 0.68 for SIFT and 0.54 to 0.61 for ABCD) while we note a slight decrease in the macro averaged precision (0.44 vs. 0.36 for SIFT; 0.37 vs. 0.34 for ABCD), recall (0.42 vs. 0.37 for SIFT; 0.40 vs. 0.37 for ABCD), and F1 (0.36 vs. 0.33 for SIFT; 0.34 vs. 0.32 for ABCD)[27].

Granularity	Approach	P_{micro}	P_{macro}	R_{micro}	R_{macro}	$F1_{micro}$	$F1_{macro}$	Acc.
Coarse	SIFT	0.72	0.58	0.72	0.63	0.72	0.58	0.72
	ABCD	0.67	0.54	0.67	0.64	0.67	0.55	0.67
Fine	SIFT	0.68	0.36	0.68	0.37	0.68	0.33	0.68
	ABCD	0.61	0.34	0.61	0.37	0.61	0.32	0.61

TABLE 8.9: Evaluation of the Prediction Task for Coarse and Fine Categories including the NOFRAME class on the Second Evaluation Set

Interestingly, also for the fine-grained prediction, we measure an accuracy of over 0.68 and 0.61 respectively which is certainly way better than we expected, referring to the (low) baseline scores from the original test set. There, we reported an accuracy for the fine-grained classification of 0.30 which even would have been to be diminished by the factor 0.77 (accuracy for the baseline on the binary task), thus resulting in a poor accuracy of 0.23.

Prediction on Frameslices

We finish this section of the quantitative benchmark by investigating the potential distinctiveness of the approaches which we compared with regard to the "core of the frames". These are the textual passages which were annotated in the original data set by the annotators to clearly depict which part of the text (or in our case the paragraph as unit of analysis) was the trigger to annotate the frame. Please note that this task

[27] Again, this is a natural outcome if the additional class to predict is rather big which increases the accuracy when predicting with approximately the same quality as for the other classes.

is of partially artificial nature; while it is completely reasonable to try to predict the category given a slice of text, we would not get any slice of text in the first place when we apply the framing detection approach to the raw text of media articles[28].

The main purpose of these evaluations is to estimate how the approaches perform in comparison to the pure task of classifying a whole paragraph. This means, if we could access the parts of the texts which lead to the annotation of a frame, how well would the categories be separable. Naturally, there is no NOFRAME category for this data set, since we only have the textual passages for cases where there are actually frames[29].

Approach	Class	Prec.	Rec.	F1	Number of Instances
Baseline	Input_Legitimacy	0.84	0.84	0.84	298
	Output_Legitimacy	0.84	0.88	0.86	385
	Throughput_Legitimacy	0.51	0.36	0.42	69
	Micro avg.	0.82	0.82	0.82	752
	Macro avg.	0.73	0.70	0.71	752
SIFT	Input_Legitimacy	0.81	0.83	0.82	298
	Output_Legitimacy	0.85	0.88	0.87	385
	Throughput_Legitimacy	0.65	0.46	0.54	69
	Micro avg.	0.82	0.82	0.82	752
	Macro avg.	0.77	0.73	0.74	752
ABCD	Input_Legitimacy	0.76	0.72	0.74	298
	Output_Legitimacy	0.83	0.80	0.81	385
	Throughput_Legitimacy	0.42	0.58	0.48	69
	Micro avg.	0.75	0.75	0.75	752
	Macro avg.	0.67	0.70	0.68	752

TABLE 8.10: Evaluation of the Prediction Task for Coarse Categories on the Original Data Set of Frameslices

In Table 8.10 we report the results for the coarse-grained case. In this case, we refer to a test set of 752 slices of text from German news media articles. Since we do this evaluation on the original data set, we include again a baseline.

[28]In an early stage of the project, this version was also considered, using a sequence labeler and a downstream classifier for the extracted passages. However, due to the bad performance of the sequence labeler, this scheme has been discarded.

[29]Of course, one could argue that we could have sampled arbitrarily textual passages from paragraphs for which we had the label NOFRAME; but this seemed to introduce too much noise without providing many new insights after a first empirical test stage.

When we investigate the results of the baseline and compare it to the results of the baseline from Table 8.4 where we predicted on the level of a given paragraph, we become quickly aware of the fact that for this special task, the baseline is a competitive approach. With an accuracy of 0.82 and macro averaged scores all above 0.7 the baseline performs remarkably well. While the ABCD approach is not improving over the baseline, the SIFT approach still delivers the best results. With the same accuracy (0.82) as the baseline, the SIFT approach shows slightly better macro averaged precision, recall, and F1 scores of 0.77, 0.73, and 0.74. While the ABCD approach is outperformed by the baseline, it still displays a decent performance (accuracy of 0.75) and also handles the small class (*Throughput_Legitimacy*) better than the baseline in terms of recall.

Overall, we note that we get high values in the evaluation of this task—also for the baseline. This leads us to two assumptions: Firstly, given the much lower performance for the same differentiation task but applied on the whole paragraph (the baseline's accuracy was 0.48), we consider finding this textual passage (or the signal) the hardest part of the framing detection and classification task. More precisely, if we already knew the exact textual passages, a mere classification along the given categories is not the big challenge. Rather, the detection of those passages is the difficult part[30]. Secondly, if the prediction of the frame class is done with such a high accuracy, chances are good, that the underlying concept (the frames) which we try to detect and classify, is actually at least distinguishable and probably also detectable.

A similar scheme is present when we look at the fine-grained classification of the textual passages in Table 8.11. Also here, the baseline outperforms the ABCD approach and even the normal SIFT approach in terms of precision. Since a basic text classification approach yielded such excellent results, we additionally created a slightly modified version of the SIFT classifier. We included a simple text classifier into the ensemble instead of the NaïveBayes classifier in order to benefit from the apparently successful Bag-of-Words representation (on which the baseline relies). We named this version SIFT_{TC}, and we see in Table 8.11 that this approach then in turn clearly outperforms the baseline with an accuracy of 0.69 and a macro averaged F1 score of 0.58 for the fine-grained labeling task.

Finally, we show a detailed evaluation of the fine-grained classification of the textual passages to illustrate the differences in performance between the SIFT_{TC} approach and the ABCD approach. We mention again, that the ABCD approach and its respective

[30] And additionally it should be mentioned, that these passages alone do not trigger a frame; it is also dependent on the context they appear in. In fact, the interplay between these factors and the strongly locally bounded information make the task so intricate. I.e., while a frame is a kind of latent variable considering the style and content of the text, its manifestation is distinctively separable.

	P_{micro}	P_{macro}	R_{micro}	R_{macro}	$F1_{micro}$	$F1_{macro}$	Acc
Baseline	0.63	0.61	0.63	0.49	0.63	0.52	0.63
SIFT	0.62	0.53	0.62	0.52	0.62	0.52	0.62
$SIFT_{TC}$	0.69	0.65	0.69	0.56	0.69	0.58	0.69
ABCD	0.53	0.44	0.53	0.47	0.53	0.45	0.53

TABLE 8.11: Evaluation of the Prediction Task for Fine Categories on the Original Data Set of Frameslices

heuristics are not learned in a data-driven fashion and therefore offer ample room for improvement.

Approach	Class	Prec.	Rec.	F1	Number of Instances
$SIFT_{TC}$	Accountability	0.45	0.38	0.42	13
	Deliberation	0.44	0.18	0.26	22
	Efficacy	0.69	0.88	0.77	268
	Efficiency	0.69	0.37	0.48	117
	Epistemic	0.71	0.81	0.76	27
	Legality	0.73	0.41	0.52	27
	Participation	0.76	0.68	0.71	37
	Representation	0.77	0.85	0.81	131
	Stakeholder	0.62	0.63	0.63	81
	Transparency	0.60	0.41	0.49	29
	Micro avg.	0.69	0.69	0.69	752
	Macro avg.	0.65	0.56	0.58	752
ABCD	Accountability	0.10	0.15	0.12	13
	Deliberation	0.12	0.18	0.15	22
	Efficacy	0.60	0.66	0.63	268
	Efficiency	0.36	0.24	0.29	117
	Epistemic	0.66	0.85	0.74	27
	Legality	0.45	0.63	0.52	27
	Participation	0.53	0.49	0.51	37
	Representation	0.63	0.61	0.62	131
	Stakeholder	0.68	0.42	0.52	81
	Transparency	0.31	0.45	0.37	29
	Micro avg.	0.53	0.53	0.53	752
	Macro avg.	0.44	0.47	0.45	752

TABLE 8.12: Detailed Evaluation of the Prediction Task for Fine Categories on the Original Data Set of Frameslices for SIFT and ABCD

When we look at the specific classes, we see that the somewhat simplistic heuristics of

the ABCD approach alone did not cope with all levels of skewness from the data. *Accountability* and *Deliberation* are still poorly captured and require further improvement. On the other hand, we get an idea of the possible improvement if we look at the performance from the SIFT_{TC} approach that benefits fully from all axes of patterns united in the ensemble (Bag-of-Words text classification as well as (crafted) lexical and semantic (similarity) features). A promising insight here is that not all smaller classes suffer from poor representation in the given approaches. Especially the SIFT_{TC} approach delivers good scores also for smaller classes like *Epistemic* and *Participation*. While we still see a slightly better performance for the well represented classes, the gap is clearly not as big as in the case of the fine-grained frame categories prediction on the paragraph-level which suffered from low macro average scores stemming from several classes with weak performance (see Table 8.5).

8.3 General Remarks

In this section we intend to summarize several insights and conclusions that we draw from the comparison in the empirical quantitative benchmark.

- The double data skew is problematic, especially for the baseline approach.
- The decomposition into a two-level task—a detection task followed by a classification task—appears to be feasible to partially tackle this problem. Additionally, this points to the general underlying rationale of frames of legitimacy in the sense that we may subsume them into a generic FRAME category.
- During the detection task, we reach a better recall for the frames by informed downsampling of the residual class.
- The prediction of the fine-grained categories is more difficult than the coarse-grained categories.
- When we look at the results of the proposed approaches (SIFT and ABCD), we note a substantial gain in performance in comparison with the baseline.
- The lexical resources that the proposed approaches finally rely on have proven to be useful for the task at hand
- The comparison of the analogous evaluation settings on a secondary evaluation set revealed notable differences: the results were consistently better for the second set

- The classification of the textual passages ("core of the frames") alone is not the hard part of the task. However, since they are not available in the normal application scenario, the approaches must cope with the prediction on units (paragraphs) which also contain a considerable amount of text that do not relate to the frames.

8.4 Chapter Summary

In this chapter, we have given a thorough overview on different approaches which were applied to the task of framing detection. More precisely, we reported on the results of the automated detection and classification of frames of legitimacy for new forms of governance in media articles.

We especially evaluated approaches based on the lexical resources that we have created for this purpose (see also Section 6.4.2). More concretely, we evaluated the performance of the framing detection component from SIFT (cf. Wüest et al. (2017)) and compared it to the ABCD approach proposed in this work as well as to a baseline relying on pure text classification techniques. The rationale behind this thorough evaluation is to study the improvements over the baseline and compare them on different levels of the classification in terms of granularity (binary, coarse-grained, and fine-grained).

We carried out this comparison on three different settings. Firstly, we compared the performance on a held-out test set from the originally annotated data set. Secondly, we applied the SIFT approach and the ABCD approach to a secondary evaluation set. Lastly, we also compared the different methods on the task to differentiate between textual passages alone which were annotated as the "the core of the frame". While these textual passages are normally not available for the automated framing analysis, we used them as another axis of insight to the framing analysis as a whole. With this comparison we also learned that detecting the frames in the text is the more difficult part since it requires attentiveness to latent information that is strongly locally bounded.

In all settings the baseline was outperformed, partially by a large margin. Also, the problem of low recall for the classes of interest (i.e., the frames) caused by the double data skew was largely tackled.

The comparison also showed that in general the SIFT classifier performs best in this task. This is an expected outcome as the SIFT approach was especially developed for this purpose and was partially strongly tuned to tackle the specific problems. On the other hand, the applied ABCD approaches made only basic usage of the annotated data. Since there is no inherent restriction to include more information from labeled data, there is ample room for improvement for the ABCD approach for such a task,

especially by improving the prediction through learning the target mapping function more in a data-driven fashion.

Overall, most of the occurring problems were addressed on an at least satisfactory and sometimes promising level and the derived lexical resources have shown to be useful and robust.

9

Discussion

> "A poor craftsman blames his tools."
> — English proverb

```
In [216]: analogy(a="Empirie", b="Ergebnis",
x="Reflexion", y=None, model_given=model, verbose=True)

'Empirie' is to 'Ergebnis' as 'Reflexion' is to 'Resultat'
Out[216]:
[('Resultat', 0.5886368155479431),
 ('Ereignis', 0.38496237993240356),
 ('Bericht', 0.37854281067848206),
 ('verlauft', 0.37230002880096436),
 ('Zustand', 0.36980128288269043),
 ('Nachdenken', 0.36862725019454956),
 ('Befund', 0.36585038900375366),
 ('Entscheidung', 0.3656880259513855),
 ('Fazit', 0.36102691292762756),
 ('Zwischenbilanz', 0.3586937487125397)]
```

In this chapter, we discuss several outcomes from the empirical experiments we have reported on in the Chapters 6, 7, and 8 and relate them to the desiderata we derived in the beginning.

9.1 On the Lexicon Induction

We have given several examples how to derive lexical resources with increasing complexity in Chapter 6. In doing so, we aimed at illustrating the productiveness and adaptability of the lexicon induction module. Given the results from the Chapters 7 and 8 where the lexical resources were applied to the problems of classification and framing detection, we consider the given approach as a valuable alternative to standard techniques.

If we go back to the list of desiderata that was developed in Chapter 2, we believe that this module works well as the main link to the established methodology of dictionary-based content analysis. We aimed at tackling the problem that off-the-shelf dictionaries (i.e., lexical resources) often only partially match the requirements of the desired application and need to be carefully adapted, leading to an increase in manual work. Therefore we have proposed a lexicon induction method that is highly adaptable for any of the intended main usages, be it creation, extension, or adaptation. We leverage computed resources from distributional semantics (embeddings) which allow us to generalize a concept (in its simple manifest form a list of words linked to the respective concept) "externally", i.e., with the help of the underlying generic semantic model of language.

One of the overarching goals of this layer of the application was to carry out a technology transfer to the target domain. Although there are uncountable alternative ways to automatically derive lexical resources, we consider the proposed approach to be a valid contribution, mainly due to its versatility as well as the integration of *a priori* knowledge combined with the possibility for interactive usage.

The choice to follow the methodology of dictionary-based approaches for automated content analysis also leads to, at least, partial fulfillment of two other aspects of our catalogue of desiderata. First, the approach allows for inspection of the resource that we would like to apply for the content analysis. Words are a currency that we human beings and speakers of language understand , in contrast to vectors or bare models using the multidimensional vector space. Second, lists of terms are easily edited and adapted in an arbitrary framework—even simple text editors suffice for this step. The goal to make the resources re-usable for other approaches than the proposed one is therefore easily achieved. This aspect also addresses the desired modularity and adaptability.

Additionally, we would like to emphasize two further properties which are linked to the proposed way of applying the resources. First, all the lexicons do not have to be mutually exclusive. This is especially interesting with respect to the possible recombination of lexical resources, or for the representation of conceptual ambiguities between involved categories or classes of interest.

Second, the requirement for a lexicon to be as exhaustive as possible is alleviated. The quantization of the lexicons into n concept detectors—artificial points in the vector space representing large portions of the lexicon—leads to a generalization during application since we rely on similarity in the embedding space in contrast to string matching methods[1].

As a last point, we would like to mention that the approach offers the possibility to profit from annotated data. For example, in Chapter 7, we used the annotated examples from the document classification to derive a "core of the concept" (here only the words that occur uncommonly often in that class) which we then expand with the help of the lexicon induction process. In such a way, the approach enables the researcher to benefit from knowledge or annotated data, or even both at the same time. This is key in cases where either data sets are not available in quantities which allow for purely supervised machine learning based solutions, or where we specifically rely on the competitive performance for all categories, no matter how small the data for it may be in relation to other categories.

9.2 On the Document Classification

In Chapter 7 we thoroughly investigated the application of the derived lexical resources for a typical task in computational linguistics which also matches many application scenarios in social sciences: document classification. More precisely, we scrutinized the case of skewed distributions on small to medium-sized data sets.

The problem of class imbalance is in itself an interesting problem and is tackled in many different ways (see Haixiang et al. (2017)). We propose in this work to tackle this problem through external generalization, i.e., we extend the respective lexical resources which represent each class in a fashion so that we counteract the lacking variety of the data points of the underrepresented classes. More precisely, we perform a lexicon induction for each class. We have demonstrated that, in particular, the recognition of the small classes could be improved in comparison with standard supervised learning approaches, thus leading to overall better macro scores (specifically recall).

In order to test the hypothesis that the application of the lexical resources on top of an embedding backed modeling will help to mitigate the problem of coverage of vocabulary, we also compared the approach to a simpler version, where we use look-up based features. Also for this comparison, we have shown, in most cases, an improvement in performance which points to the competitiveness of the presented approach.

[1]Nevertheless, a combination of (sub-)string matching methods and concept detector similarity measurement is not prohibited in any way and has proven useful for the framing detection task from Chapter 8

We propose to re-embed the lexicon and cluster it to use the resulting n centroids as concept detectors to measure their signal given the unit of analysis. We consider the sentence-level to be more apt for similarity-based comparisons of this kind—rather than compressing a whole document into a point in the semantic space. This leads us to propose a (simple) method to aggregate the sentence-based measurement to the document-level.

Nevertheless, as we have always emphasized the importance of modularity, different sentence embeddings methods could be applied instead of the simple averaging method for word embeddings (coupled with filtering based on part-of-speech)—as long as they allow for a comparison with the concept detectors. If this kind of comparison is not realizable due to the nature of the sentence embedding, this part of the approach would have to be adapted.

Furthermore, we also suggest that more advanced methods to propagate the sentence-wise measurements could improve the results regarding the prediction of the document label. Obviously, the projection of the aggregated similarity scores to a document label may be learned in a supervised manner instead of heuristically predicting the highest scoring class as we did with the ABCD approach (see Chapter 5). However, in this work we focused also on simplicity and therefore compared a simple heuristic unsupervised classifier to the other classifier methods. If we tune the two parameters of the classifier, we are even able to improve further the results for the specific cases at hand.

Since all of the classes are independently measured via their concept detectors, the two parameters (similarity threshold and n-best) allow for concentrating on the sentences that carry enough signal to be meaningful for the prediction in a simple manner. In other words, if the signal we measure is too low for all classes, we just ignore the sentence. If we interpret the concept detectors as high-level features, it becomes clear that a cut-off of irrelevant noisy signal is helpful for the prediction.

When we further stick to the perspective of the sentence-level modeling, we may also link to some of the overarching goals of the implementation. In addition to the performance gain through the focus on signal-bearing sentences, the approach also offers possibilities for an in-depth analysis of the prediction. For each sentence, we have a comparison to all concept detectors. Since we know by inspection which sub-concepts are represented by the concept detectors, we receive a fine-grained analysis on the sentence-level and for each class (modeled as n concept detectors). In other words, for each prediction, we are for instance able to deliver a back reference to the passages in the text, where the signal was the strongest (or weakest) for each respective class.

This in turn allows also for a more complex modeling of document classification where sub-units of the documents are taken into account, i.e., paragraphs, headlines, teasers, and so on. In combination with the ideas of zoning (cf. Teufel (1999) for scientific texts), this idea seems to be promising also for more detailed media content analysis which investigates and compares different sub-units of documents or articles.

With the opportunities for a transparent fine-grained analysis, we hope to hand over a powerful tool to the researchers in this domain. Since we have avoided black box modeling, we allow for a high level of control and inspectability that also permits fine-tuning of the applied resources.

9.3 On the Framing Detection

In Chapter 8 we have reported on results for the task of framing detection. This work—including the whole process of data acquisition, sampling, and annotation—was conducted in the NCCR democracy research project and is thus the result of intensive collaboration with many other researchers. We gladly took up the opportunity to report with a focus on the development of the method for automated content analysis in more detail as part of this thesis.

In the beginning of the project, the task was planned to be solved with a supervised machine learning approach based on standard techniques from the domain of document classification. As it turned out, the results that were achieved in this way were not satisfying. Because of the challenging general set-up (fine-grained target-specific framing analysis in three languages in media outlets from four countries over a time period of ten years), we had to adapt to the redefined requirements.

First of all, the unit of analysis was changed from the document to the paragraph since the prediction on the document-level turned out to be too coarse for the intended downstream analyses[2]. Because of this change in the unit of analysis, we ended up with a data distribution which we referred to as *double data skew*. On the one hand, many more paragraphs in the corpus of media articles did not contain any frame. This means that the residual class was highly overrepresented. On the other hand, there was also a heavy skew in the distribution over the classes (in this case different subtypes of legitimacy frames).

The problem of imbalanced data and the simultaneous over-representation of the residual class (which was not defined in any other way than by the absence of the frames) lead

[2]As mentioned on several occasions, the frames should be linkable to specific named entities and hence not all occurring frames in a text should be deemed to be related to all occurring named entities in the text

us to design the solution as a stacked classification, i.e., to predict first the presence or absence of a frame in general and subsequently classify the frame. While the results achieved with standard techniques for the first binary prediction were acceptable, it turned out that these methods were insufficient for the more fine-grained differentiation. Especially for the frame classes that were underrepresented in the data sets, the recall values were often too low (let aside the results from the attempt to predict the fine-grained frames with a flattened class modeling).

In order to counteract this, we had to develop a method to cope with the double data skew. At this point, the link to lexicon induction techniques was made in order to externalize the generalization. This led to the development of the SIFT approach which incorporated elements from the proposed approach from this work (ABCD). But the SIFT approach additionally relied on several tailored optimization methods to tackle the problems.

The different benchmarks all show substantial improvements over the baseline, especially by the SIFT approach. But also the more generic ABCD approach—without the specific tweaks for the case at hand—outperformed the baseline in most cases[3]. Nevertheless, the gap in performance between SIFT and ABCD might point to the potential to further improve the proposed approach from this work when applied in hybrid combination with other machine learning methods.

Although the SIFT approach may be considered as a successful adaptation of the generic ABCD approach—in the sense that it makes also heavy usage of annotated data—one has to keep in mind that it is much more complicated in design and is also restricted to the prediction of the label, in contrast to the in-detail analysis which the ABCD approach allows for.

Another detail we would like to emphasize is the difference with regard to achieved scores when we compare the first and the second data set. It turned out that both variants (SIFT and ABCD) performed notably better on the second data set. While it remains unrevealed if this is due to the sampling of the data, the different annotators, or both, we interpret this as a stong indicator to take quantitative evaluation benchmarks with a grain of salt. We also highly recommend to always accompany an automated media content analysis with manual annotation, at least for evaluation purposes.

In the last experiment we have reported on the prediction of the frame class for spans of text (to which we had access because of the explicit annotation method) for a specific in-depth evaluation. Interestingly, it turned out that this part of the prediction was not

[3]This is notable because it assesses the performance of an unsupervised approach against classifiers that are learned in a supervised fashion.

causing the problems—for none of the compared approaches. This indicates again, that the detection of the signal in the text is the hardest part and not the classification of those text spans. However, it is important to note that those spans which point to the core of the frame are normally not known. As a consequence, the empirical outcomes are not directly applicable to the original task but nevertheless elucidate which part of the task is presumably more difficult.

To sum up, we consider the proposed solution as a valuable contribution for the further development of methods for this task. Although we reached an encouraging level of performance in the end (especially on the second evaluation set), we still suggest further investigation into the persisting problems for automated framing analysis. For the automated content analysis, we need to develop alternative approaches which may profit from the insights we described in this work—or, in the best case, are even built upon the pillars of the given solution.

9.4 On the Approach in General

If we look more holistically at the proposed approach, we like to point out several of the central insights from its application to the different tasks.

First, we believe that the inclusion of the induction of lexical resources makes it versatile. Trivially, this is a mere consequence of the versatility of the lexicon induction approach. If we are able to aptly derive lexical resources for concepts that we want to model in our content analysis, the proposed approach allows for a rich variation of application, ranging from document classification to much more fine-grained analysis of smaller parts and specific zones within a text.

Second, since most parts of the pipeline are exchangeable[4]—for example, the lexical resources may have different origins and are completely independent from the classification—the approach can also be conceived as flexible and malleable. More precisely, we identify at least the following core elements as modules that are replaceable by any appropriate alternative:

- The embedding model may be freely replaced, either for lexicon induction or re-embedding and prediction.

[4]In fact, the only hard restriction is that the embedding model with which we calculate the concept detectors from the lexical resources must be the same as the one that is used to model the sentences of the text.

- The derivation of the lexical resources may be combined with any other method to identify the "core of the concept", be it solely data-driven, solely knowledge-driven, or any combination of the two.
- The clustering method to re-embed and quantize the lexicon may be chosen to fulfill specific requirements. For simplicity we employed k-means clustering and even kept the number of clusters constant for most experiments. However, we recommend adapting the number of clusters to the size of the lexical resources. The inspection of the vicinities of the centroids allows for control over the coverage and focus of the concept detectors.
- The proposed classification layer of the approach is geared towards transparency and hence simplicity. While the results of the heuristic unsupervised classifier are competitive, the usage of the lexical resources via the concept detectors is not restricted to this kind of usage by any means. Any combination with supervised machine learning models is imaginable, given enough annotated data is available.

To sum up, we conclude that the proposed approach—including all its modules—works encouragingly well for a range of applications. This is promising since we were able to stick to the given guidelines and desiderata which have been formulated, including the perspective of social science on media content analysis.

9.5 A Word on the Empirical Justification

On the one hand, we have designed the approach to be interactive in the sense that the user is given the opportunity to heavily influence the lexicon induction. Additionally, there are multiple points where the user is given the possibility to inspect the intermediate state of the resource (the lexicon and the concept detectors) and reconsider prior decisions in order to iteratively reach the desired outcome. On the other hand, we have presented mainly "scripted" examples in the empirical part on the lexicon induction. Those examples required no interaction at all, but also did not profit from the potential gain of human intervention. This is also due to the difficulties of creating reproducible results for experiments when allowing for human interaction.

Nevertheless, to emphasize the flexibility of the lexicon induction process, we presented a broad mixture of experiments. Those are partly of an explanatory and elucidating nature and partly serve a rather descriptive purpose[5]. The empirical evaluations of the

[5] In the sense that we explain how we created the resources that we applied in the document classification and framing detection as extrinsic tests.

lexical resources (and its application) on pre-annotated data from different domains with different domain challenges points to the versatility and adaptability of the approach. However, in order to not stress the scope of this work, we restrained from multi-labeling cases, although the model is exactly doing this under the hood.

Although the performance is remarkably good and robust, one of the lessons we draw from the broad range of experiments is that empirical verification is essential for any new domain. It is our duty as users of automated analyses to be skeptic and to demand for systematic and rigorous testing before applying our approaches to large amounts of data.

9.6 Limitations of the Approach

In this section, we will attempt to bring together the most important points that show the limitations of the approach. While the overall evaluation results in a promising image, we believe that it is nevertheless necessary and helpful to shed light on the more problematic aspects, too.

First and foremost, the presented approach aims at detecting concepts (defined by lexical resources). This leads to a strong focus on the subject or the theme of the textual units that we analyze. To be more precise, the proposed approach is most suitable for cases where we are mainly interested in what the texts are *about*.

While we partly overcome the problem of coverage (given the vocabulary of the lexicon) by modeling the content and the lexicon based on an embedding, we still focus on the axes of vocabulary and meaning. If the goal is to analyze other aspects which are more connected to the form and style (e.g., the use of pronouns, the use of intensifiers), we recommend applying other techniques or other features which do not rely on a word embedding. The main reason is that words or terms which occur in many different contexts (such as prepositions, pronouns, etc.) do not have a very distinctive representation in the embedding (cf. Lau and Baldwin (2016), Adi et al. (2016)).

The same recommendation holds true if the phenomenon under investigation is connected to idiomatic or metaphorical usage of language. This kind of non-compositional expressions are not accessible to the simplistic construction of a sentence embedding[6]. Especially when following the conception of framing analysis from Entman (1993, p. 52) where he states explicitly that the "text contains frames, which are manifested by the presence or absence of certain keywords, stock phrases, stereotyped images, sources

[6]Of course, one could for instance model idioms separately in the embedding. But, as a consequence, this requires also a non-trivial stage of preprocessing of the textual unit we would like to analyze.

of information, and sentences that provide thematically reinforcing clusters of facts or judgments". One should be attentive that these stock phrases and keywords are detected and aptly modeled in an automated framing analysis.

Second, we leverage the semantic representation of terms through an embedding. As a consequence, this leads to a strong dependency on this representation. Therefore, we will also have to face all kinds of issues which are connected to this modeling, such as a poor or even completely lacking representation of rare words (see Luong et al. (2013) or Zesch and Gurevych (2006) for German), domain vocabulary, or introduced bias (see Bolukbasi et al. (2016) and Caliskan et al. (2017)).

Fortunately, several methods to counteract such problems have been developed in recent years. Subword-based modeling (Joulin et al., 2016) provides us with a fallback for unknown words. Retrofitting techniques (Faruqui et al., 2016) may be used to shift the meaning of badly represented terms, or to manipulate the embedding in general in order to adapt the relations to one's expectations. Furthermore, there are attempts to solve the problem of unwanted bias of the model (cf. Speer et al. (2017)).

However, these are all sophisticated methods which require a substantial amount of knowledge which may not be accessible for the social scientist who is the intended user. It is therefore meritorious when pre-computed models are made available for which a whole set of optimization strategies have been applied (cf. Mikolov et al. (2018), or Speer et al. (2017)).

Third, although we consider the approach to mainly follow the guidelines that we have derived for its development, the question may arise whether simplicity has had enough influence on the approach. While the machinery for classification tasks in general relies on only two parameters, the (full) potential for optimization will be mostly only accessible for technology-savvy users. This is an outcome of the trade-off between simplicity and flexibility (or in this case also complexity). At a certain point, the former cannot be forced without diminishing the latter. Nevertheless, we think that two parameters are acceptable in terms of complexity—especially given their clear and transparent influence on the algorithm. And, as a last point, the performance with default parameters is robust and delivers good results for a wide range of applications.

10

Open Questions and Future Work

> "All generalizations are false, including this one."
> — Mark Twain

```
In [215]: analogy(a="Empirie", b="Ergebnis",
x=None, y="Reflexion", model_given=model, verbose=True)
'Empirie' is to 'Ergebnis' as 'Subjektivität' is to 'Reflexion'
Out[215]:
[('Subjektivität', 0.5636058449745178),
 ('Erzählen', 0.5436885952949524),
 ('Imagination', 0.5313553810119629),
 ('philosophisch', 0.5282355546951294),
 ('Poetik', 0.5245558023452759),
 ('erzählerisch', 0.5233293175697327),
 ('reflexiv', 0.5212591886520386),
 ('metaphysisch', 0.5201999545097351),
 ('Dialektik', 0.5159838199615479),
 ('dialektisch', 0.5100311636924744)]
```

In this chapter, we discuss some of the questions which remain open since their treatment goes beyond the scope of this work.

First, we will briefly address the question on representation. In this work, we have opted for a modeling that creates a representation of a sentence as a singular point.

In recent research, there have been many attempts to learn such a representation for a sentence incorporating also its inner structure—which is clearly more advanced than the simplistic approach we have followed in this work, where we average over a subset of words pertaining to a particular part-of-speech (mainly nouns and adjectives).

Second, we will sketch the cross-lingual possibilities for approaches that rely on a modeling based on embeddings and use lexical-semantic resources as our approach does. There are multiple angles from which one may leverage cross-lingual resources such as Conceptnet (Speer et al., 2017), the aligned `fastText` models (Joulin et al., 2018) and other approaches which merge embeddings across languages (for example Artetxe et al. (2017) and Artetxe et al. (2018)).

Third, we will take up this thought of merging different embeddings but switch from the cross-lingual scenario to a domain-adaptation perspective. There, we consider cases where domain-specific semantics may lack in a general semantic model or be deviant.

Fourth, we will reflect on the possibilities of the granularity and diversity of the concepts for which we create the lexical-semantic resources. On the one hand, we may create them in an application-oriented way to solve a specific task, like in Chapter 7 and 8. On the other hand, since these concepts have proven to be useful, why not create a plethora of them based on basic categorical information available from wikipedia? In combination with a fast implementation which checks for the occurrence of such a catalogue of concepts, such an implementation could turn out to be widely applicable for classification scenarios.

Fifth, we would also argue for hybrid approaches which combine syntactical information from parse trees and the power of the lexical-semantic resources.

Sixth, we also discuss briefly the widening of the window of information from the sentence to larger units of information. In one way, we have done exactly this by predicting a label for a whole document based on sentence-level modeling and prediction (Chapter 7). Also the task of framing detection and prediction we reported on in Chapter 8 is tackled using the sentence-wide representation aggregated to the unit of the paragraph. In this section, we would like to mention a couple of alternative approaches which go beyond the focus of this work but which potentially foster its usefulness for other scenarios.

10.1 What to Embed

In the extrinsic empirical evaluation for the lexical resources we have modeled the content on the sentence-level. More precisely, we used a representation of the sentence by a

simple averaging method, applied on words of a specific part-of-speech set (i.e., nouns and adjectives). Subsequently, we compared this representation of the sentence with the concept detectors which represent the lexical resources in a quantized way. This comparison has proven to be useful (as the results from Chapter 7 suggest) and efficient—also with a large number of concept detectors—as the similarity calculation is vectorized.

Of course, one could also do this comparison based on word-level but there are at least three points which favor the sentence-level approach. First, in terms of efficiency, the sentence-level is as many times less computationally expensive as the ratio of the number of terms to be embedded to the number of sentences is. Second, because of the additive compositionality properties of the embedding (see (Mikolov et al., 2013b, p. 7)), we also get a disambiguation of the word senses through the context, i.e., the other words in the sentence that are used for its summative representation.

Third, recent research suggests that the sentence may be the reasonable level up to which a representation of units as vector points is feasible (cf. Conneau et al. (2018)). In other words, the representation of larger units (e.g., paragraphs or whole documents) tends to be connected with an inherent loss of information, although this is the goal of approaches such as `doc2vec` (Le and Mikolov, 2014)[1].

Alternatively, we may use a data-driven method to decide up to what length and combination a term (or token) might be suitable to embed as it was proposed by Gyllensten et al. (2019). In this case, based on the idea of byte-pair encoding as introduced by Sennrich et al. (2016), the authors recursively derive the units which are to be embedded in a data-driven way, i.e., they find the strings of characters (including white space and punctuation) which should be considered as tokens. Since the resulting "tokens" are hence determined according to their patterns of occurrence, these units align well with the idea the embedding should contain semantically motivated units[2].

10.2 Cross-lingual Scenarios

While we could of course translate the lexicons term-wise to apply the translated lexical resources for applications in other languages, there is an interesting opportunity to apply

[1] It should be mentioned that the goal of these approaches is normally not to generate an embedding model to apply lexical resources on but to create a distributed representation of units such as paragraphs, or documents so that they are comparable, i.e., the similarity between them is quantifiable

[2] Note that the inclusion of multi-word terms was also tackled by earlier approaches such as `word2vec` (Mikolov et al., 2013b) but more as a part of the preprocessing, namely, by calculating collocations in a data-driven way.

them with minimal additional transfer costs and leave out the translation step[3].

One of the advantages we have not devoted much attention to, is that through advances in research on cross-lingual embeddings the presented approach is only one step away from a cross-lingual application. The idea of cross-lingual embeddings is that one embeds the model of distributional semantics of at least two languages in a *shared* space, so that we have an alignment in meaning. Several approaches to derive such multilingual embeddings have been described and pre-trained models have been made available (see Speer et al. (2017), Lample et al. (2018), Artetxe et al. (2017), Joulin et al. (2018), Artetxe et al. (2018)).

While we have demonstrated that classification on different tasks is performed accurately with the concept detectors, the question is how aptly they represent the same sub-concepts (and hence the concept from the lexicon as such) in other languages if we use the *same* centroids for another language. We will not carry out any experiments geared at an extrinsic evaluation of cross-lingual application here, but we will give two examples to illustrate the possible application of the lexical resources to texts in a different language.

For the first example, we look again at the centroids to detect frames of transparency (see Section 6.4.2.1 and Table 6.24). In order to transfer the concept detectors, we use the cross-lingual embeddings from ConceptNet Numberbatch[4] (Speer et al., 2017). Since we use another embedding than in the original case from Chapter 8, we have to re-embed and cluster the lexicon in the ConceptNet embedding in the first place.

In Table 10.1 we see the nearest neighbors of the resulting centroids or concept detectors for the German lexicon[5]. In Table 10.2 we report the nearest neighbors from the same centroids in the aligned embedded space for French, i.e., the centroid order is the same. As we observe, there is a relatively large overlap in what the concept detectors capture in both languages. We emphasize that the centroids are computed in the aligned space, and thus no further adaptation is required.

The choice of the multilingual embedding is not restricted by any factor, so we could also use any other aligned embeddings like the ones from `fastText` or the MUSE embeddings[6]. Of course, this cross-lingual application comes at the cost that one relies on

[3]The seemingly easy approach to apply term-wise translation is additionally leading rapidly to the well known problem arising from the multitude of possible translations for single terms (see for example Vicente and Saralegi (2016)).

[4]retrivable at `https://github.com/commonsense/conceptnet-numberbatch`. We use the vectors from the multilingual embedding version 17.04.

[5]Note that only 212 words from 329 in the lexicon existed in the ConceptNet embedding for German, hence the slight difference in the clustered re-embedding

[6]Retrievable from `https://fasttext.cc/docs/en/aligned-vectors.html` and `https://github.com/facebookresearch/MUSE`

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	undurchsichtig	0.7954
2	undurchschaubar	0.7304
3	opak	0.6964
4	durchsichtig	0.6868
5	irreführung	0.6731
6	durchschaubar	0.6726
7	feuilletonistisch	0.6703
8	unehrlich	0.6640
9	unscharf	0.6603
10	verquast	0.6566

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	informieren	0.9075
2	benachrichtigen	0.8943
3	mitteilen	0.8870
4	vermelden	0.8542
5	verlautbaren	0.8537
6	bescheid_sagen	0.8435
7	melden	0.8400
8	bescheid	0.8397
9	benachrichtigung	0.8364
10	übermitteln	0.8166

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	darlegen	0.8124
2	aussagen	0.8015
3	dartun	0.7959
4	hinweisen	0.7933
5	verlautbaren	0.7905
6	prätendieren	0.7713
7	aufzeigen	0.7636
8	vorbringen	0.7627
9	besagen	0.7617
10	deuten	0.7610

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	veröffentlichung	0.8950
2	veröffentlichen	0.8899
3	veröffentlicht	0.8862
4	publizieren	0.8851
5	publikation	0.8578
6	publiziert	0.8333
7	herausgabe	0.7855
8	edieren	0.7592
9	herausgeben	0.7584
10	separatdruck	0.7134

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	geheim	0.9046
2	verhohlen	0.8949
3	verborgen	0.8945
4	verschweigen	0.8786
5	verheimlichen	0.8622
6	versteckt	0.8536
7	geheimhaltung	0.8503
8	verschwiegen	0.8238
9	heimlich	0.8229
10	verhüllt	0.8227

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	geheimnachricht	0.7470
2	geheimdokument	0.7192
3	geheimakte	0.7079
4	wahrheitsfindung	0.6696
5	findung	0.6573
6	dpa_information	0.6554
7	decouvrieren	0.6523
8	deutsch_verzeichnis	0.6481
9	demaskierung	0.6479
10	konzernanhang	0.6471

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	unterlagen	0.8334
2	dokument	0.8145
3	dokumentieren	0.8003
4	aufzeichnung	0.7869
5	daten	0.7760
6	akte	0.7610
7	bewerbungsunterlage	0.7474
8	bericht	0.7351
9	registratur	0.7270
10	paketkarte	0.7247

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	aufklären	0.8333
2	darlegen	0.8110
3	erläutern	0.8029
4	klären	0.8029
5	erklären	0.7980
6	abhandeln	0.7828
7	verdeutlichen	0.7723
8	dartun	0.7566
9	gneissen	0.7536
10	herauskristallisieren	0.7522

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	detailliert	0.9279
2	ins_detail_gehend	0.9047
3	detailreich	0.8999
4	ins_einzelne_gehend	0.8907
5	ausführlich	0.8893
6	eingehend	0.8827
7	ausführlichkeit	0.8534
8	einzelheit	0.8052
9	ins_detail_gehen	0.8037
10	detail	0.7903

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	untersuchen	0.8632
2	inquirieren	0.8519
3	nachfragen	0.8350
4	anfragen	0.8307
5	nachforschen	0.8289
6	erkundigen	0.8253
7	prüfen	0.8219
8	erkunden	0.8194
9	hinterfragen	0.8183
10	begutachten	0.8083

TABLE 10.1: 10 most similar German terms to the 10 centroids of the cluster model for the German lexicon for transparency in the semantic space of the ConceptNet model, ordered by cosine similarity

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	opaque	0.6073
2	transparent	0.5519
3	flou	0.5373
4	malhonnête	0.5244
5	voilé	0.5220
6	transparence	0.5145
7	limpide	0.5041
8	malhonnêteté	0.4943
9	translucide	0.4909
10	nébuleux	0.4875

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	informer	0.7488
2	faire_savoir	0.7419
3	faire_part	0.7327
4	renseigner	0.6751
5	notifier	0.6689
6	communiquer	0.6548
7	renseignement	0.6528
8	transmettre	0.6452
9	aviser	0.6369
10	annoncer	0.6321

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	démontrer	0.6330
2	prouver	0.6289
3	confirmer	0.6267
4	indiquer	0.5986
5	affirmer	0.5967
6	attester	0.5909
7	déclarer	0.5901
8	énoncer	0.5891
9	à_mots_couverts	0.5735
10	démontré	0.5653

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	publication	0.7946
2	publier	0.7885
3	éditer	0.6935
4	publié	0.6767
5	rendre_public	0.6607
6	parution	0.6520
7	édition	0.6455
8	publient	0.6426
9	publies	0.6364
10	publicateur	0.6353

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	secret	0.8176
2	caché	0.7819
3	cacher	0.7557
4	dissimuler	0.7428
5	dissimulé	0.7271
6	cachotterie	0.6877
7	caches	0.6850
8	en_secret	0.6835
9	tenir_sa_langue	0.6826
10	confidentiel	0.6682

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	irrévélable	0.6329
2	désannoncer	0.6274
3	absconser	0.6229
4	décolléter	0.6222
5	suréclairer	0.5971
6	sur_éclairer	0.5971
7	sous_éclairer	0.5971
8	détailleux	0.5969
9	latitant	0.5941
10	trop_perçu	0.5920

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	documents	0.7001
2	document	0.6720
3	documentation	0.6025
4	compte_rendu	0.5955
5	prendre_des_notes	0.5954
6	données	0.5933
7	documentaliste	0.5773
8	registre	0.5759
9	papiers	0.5737
10	enregistrement	0.5733

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	expliquer	0.6846
2	clarifier	0.6433
3	explicitation	0.6272
4	éclaircir	0.6149
5	comprendre	0.6000
6	expliciter	0.5789
7	démêler	0.5688
8	élucider	0.5668
9	clarifié	0.5534
10	énoncer	0.5487

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	détaillé	0.8484
2	minutieux	0.7425
3	détails	0.6886
4	détail	0.6846
5	circonstancié	0.6844
6	détailler	0.6658
7	détaillée	0.6568
8	précis	0.6498
9	minutie	0.6436
10	minutieusement	0.6419

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	examiner	0.7937
2	enquêter	0.7486
3	investiguer	0.7309
4	vérifier	0.7190
5	se_renseigner	0.7185
6	enquérir	0.7157
7	inspecter	0.7120
8	interroger	0.7014
9	questionner	0.6941
10	consulter	0.6937

TABLE 10.2: 10 most similar *French* terms to the 10 centroids of the cluster model for the *German* lexicon for transparency in the semantic space of the ConceptNet model, ordered by cosine similarity

the semantic information represented in the models.

This example does not serve to be more than a proof of concept, but it shows how it is in principle possible to apply resources derived in one language on units of analysis in other languages. It is important to emphasize that the applicability would have to be more thoroughly tested in concrete cases but this lies beyond the scope of this work. But a glance at the "translated" centroids (in fact, we are inspecting the nearest neighbors of the centroids to words from other languages) reveals a promising picture, especially given the minimal transfer costs.

As a second example, we use a lexicon of natural disasters in German and investigate the transfer of the concept detectors to French and English. The lexicon (originally derived with the lexicon induction techniques from Chapter 6 and with the embedding used in the other experiments) consists of 297 terms. We find 107 terms of the original 297 terms in the German vocabulary of the ConceptNet embeddings, which is substantially smaller than the one used for the other experiments in this work[7].

The results of the clustering process of the re-embedded German lexicon are presented in Table 10.3, where we display the vicinities (20 nearest neighbors) of the centroids of the clusters. Due to the alignment of the embeddings we just collect the nearest neighbors from each language. When we investigate the results, we see that the clustering process (this time into 5 clusters) separates the lexicon[8] in five clearly distinguishable sub-concepts.

While the first centroid represents natural disasters in general, the other centroids capture thematically focused types of natural disasters. Centroid 2 describes flooding, centroid 3 references earthquakes, centroid 4 captures storms and heavy rain or snow while centroid 5 represents drought and the water shortage. In the second and the third column we list the nearest neighbors in English and French to the *same* centroids (derived from the German lexicon). To be clear here: no further steps have been undertaken to transfer the centroids as a quantized representation of the German lexicon to the other languages. Only the semantic axis of the shared representation is used.

An interesting detail here is the erroneous *implosion_therapy*[9] in centroid 2 from the English ConceptNet model. This is a term from the psychological domain (see Astrup

[7]Note that this is also due to the differences in the automated calculation of collocation from the different models. In the original list there are 138 bigrams which were not present in the ConceptNet model.

[8]see Appendix C for a full list of terms of the lexicon.

[9]It seems that *implosive therapy* is more common in the psychological literature. However, *implosion_therapy* is the term which is present in the ConceptNet model.

(1978) for example) which is closely linked to the term *flooding*. Flooding refers to a behaviour therapy where the goal is to treat phobia and anxiety disorders[10].

	German Model	English Model	French Model
1	naturkatastrophe, katastrophe, katastrophal, verheerend, desaster, kalamität, verwüstung, unheil, naturereignis, ereignis, klimakatastrophe, verwüstend, naturphänomen, regenereignis, stufenjahr, katastrophenschutz, starkregen, super_gau, unglück, folgenschwer	catastrophe, disaster, natural_calamity, calamity, natural_disaster, catastrophic, great_disaster, cataclysm, cataclysmic, devastation, disasters, catastrophes, ravages, course_of_events, disastrous, devastating, cataclysmal, laying_waste, calamities	catastrophe, désastre, calamité, cataclysme, catastrophique, ravage, risque_naturel, cataclysmique, cata, désastreux, ravages, raz_de_marée, catastrophe_naturelle, vimaire, destruction, mégacatastrophe, anéantissement, sauver_les_meubles, dévastation, ravageur
2	hochwasser, überschwemmung, flut, überflutung, überschwemmen, blanker_hans, flutkatastrophe, sintflut, sturmflut, jahrhunderthochwasser, sturzflut, fluten, überfluten, jahrhundertflut, flutopfer, flutwelle, inundation, hochwasserschaden, springflut, sintflutartig	flood, inundation, flood_disaster, flooding, floods, freshet, deluge, landflood, flood_tide, outburst_flood, implosion_therapy, beflood, inundate, floodwaters, diluvian, flashflood, deluges, diluvial, tidal_waves, floodtide	inondation, inondations, déluge, inonder, inondable, marée_haute, ennoiement, déluges, diluvial, raz_de_marée, diluvien, inondé, submerger, ennoyage, risque_naturel, inondée, pluies, déluger, submerges, submergeons
3	erdbeben, meerbeben, seebeben, hauptbeben, erdstoss, vorbeben, nachbeben, seismisch, beben, tremor, erdbebengebiet, tsunami, erdbebensicher, seismische_woge, erdrutsch, schlammlawine, magnitude, mure, bergrutsch, bergsturz	earthquake, temblor, tremblor, terremote, seaquake, seism, earthdin, earthquakes, quake, earth_tremor, quakes, moonquake, mainshock, teletsunami, microseism, megaquake, planetquake, megatsunami, seismicity, foreshock	tremblement_de_terre, séisme, séismes, sismogène, séisme_sous_marin, microséisme, macroséisme, trémor, catastrophe_naturelle, échelle_msk, macrosismique, sismicité, tsunami, tremblement, parasismique, tsunamis, sismique, échelle_de_richter, raz_de_marée, sismographie
4	unwetter, platzregen, sturmschaden, gewitter, sturmwetter, niederschlagsereignis, starkregen, regenwetter, eisregen, sturm, wolkenbruch, schneeregen, schneegewitter, schneesturm, hagelwetter, schneetreiben, wetteraussichten, nebelwetter, regenguss, tiefschnee	winter_storm, rainy_weather, rainstorm, thunderstorm, storm, heavy_rain, snowstorm, downpour, snow_storm, thundersnow, freezing_rain, hail_storm, landphoon, tornadocane, it's_raining, thundershower, rain, meteorological_event, storms, rainfall	orage, tempête, fortunal, tempête_de_neige, pluie_torrentielle, pluie, sous_la_pluie, giboulée, météorologie, grésil, pluies, blizzard, trombe_d'_eau, tempêtes, gibouler, pluie_verglaçante, précipitations, squall, ondée, orages
5	dürreperiode, regenmangel, dürre, wasserarmut, wassermangel, aridität, wassernot, trockenheit, dürren, nahrungsmittel_knappheit, hungerkatastrophe, trockenperiode, hungersnot, trockenzeit, trockenes_wetter, austrocknung, knappheit, lebensmittelknappheit, trinkwassermangel, nahrungsmangel	drought, water_shortage, drouth, droughts, dry_weather, dry_period, dryth, dry_spell, famine, aridity, dry_season, aridness, rainlessness, absolute_drought, crop_failure, dryness, siccity, irish_famine, waterlessness, droughty	sécheresse, siccité, famine, sècheresse, xérothermophile, famines, saison_sèche, crevaison_de_faim, pénurie, disette, sous_alimentation, aridité, asséché, faim, se_tarir, faminer, inondations, faims, vaches_maigres, ombrophile

TABLE 10.3: Nearest neighbors to five centroids representing a German lexicon of natural disasters in German, English and French in a shared embedding space (ConceptNet Numberbatch)

[10]See `https://en.wikipedia.org/wiki/Flooding_(psychology)`

10.3 Bringing Different Embeddings Together

The application that we presented in this work uses a word embedding as the semantic back end. This model was calculated from a large sample of media texts (see Chapter 4). The semantics that are incorporated in this model thus rely on the content of those texts.

More precisely, while following the distributional hypothesis which states that the meaning of a word is defined through the contexts in which it occurs (see (Firth, 1957, p. 11)), the representation relies on the presence of an apt distribution of examples for the words we would like to model. This means on the other hand that if we do not have enough examples of occurrences for a given term, it (or its meaning) will not be represented well in the model. Even worse, if the number of occurrences is lower than a given threshold, the term is not represented at all—and obviously the same holds true for words we have not encountered at all in the corpus we calculate the embedding on. While this results in an out-of-vocabulary problem, there are several ways to tackle this problem (see Section 3.1.3).

But imagine the case where we need to adapt the embedding to the special semantics of a specific domain. While the out-of-vocabulary problems certainly arise for specific domains (i.e., texts from these domains contain words that do not occur in other domains), there is the second challenge, that a term has a specific meaning in this domain.

In order to adapt the modeling of such words to aptly represent them with the embedding, we recommend using one of the approaches which merge two or more embeddings and keep the meaning of the domain-specific embedding in order to get a broad vocabulary coverage combined with in-domain semantics. While these approaches were mainly developed to link semantic spaces across languages (cf. Lample et al. (2018), Joulin et al. (2018) or Artetxe et al. (2018)) there is good reason to apply the same core idea for the combination of a general semantic model and a domain-specific model. In order to do so, one only needs to identify and mask the domain-specific vocabulary and use the non-domain-specific as linkage (like the supervision-based versions of the merging approaches).

Another way is to intentionally shift the meaning in a known direction by applying the ideas of retrofitting (Faruqui et al., 2016). When it is known how the meaning should be changed (or to what the representation should be more similar), this is achieved with "nudging" the embedding of such terms in a specific direction (Faruqui et al., 2016). While the retrofitting approach was originally motivated to improve an embedding through the inclusion of semantic knowledge (synonymy, antonymy, paraphrasing), one could also use the domain-specific embedding to gradually shift terms with a domain-specific shift in meaning in a desired direction.

In general, a generic embedding will suffice as the semantic back end for the proposed approach. However, we still recommend to either train an own embedding on the textual material one wishes to operate later on if feasible, or at least to strive for inclusion of idiosyncrasies of the domain. In particular, known domain-specific discrepancies in the meaning of important terms which also occur frequently should be handled accordingly.

10.4 Generalizing Concepts

One question that remains unanswered in this work is how useful such concepts are if we step away from focused classification tasks where we craft tailored versions of such concepts so that they match the classes we want to distinguish. In other words, can we derive universally applicable concepts, let us suppose for "war", "love", "risk", "chances", "beverages", "dog breeds", or "transparency" for example with the help of wikipedia resources?

Such concepts, represented as lexicons in lists of words/terms (or the equivalent attribution of words to such concepts, for example as in the General Inquirer (Stone et al., 1966)), should be usable as a modeling background in a multitude of application scenarios[11]. This is what distinguishes them from a learned model which differentiates between classes, i.e., a model that is learned according to an objective function (for example minimization of cross-entropy loss or maximizing likelihood) given some data. Consequently, such models are normally apt to solve the task at hand (separating the classes and hence predicting the according labels) but are actually geared to fulfill the objective function following a specific optimization technique. To be clear, what makes them effective is the automated way to find a solution (or the optimal parameters for a solution) that fits the data the best. If we apply a standard Bag-of-Words modeling, some of these models produce as a side-effect something close to an ad-hoc lexicon[12]. However, these "lexicons" are built to *differentiate* between the classes and not *represent* them. Additionally, they will include any (meaningless) short-cut that is present in the data set if it not counteracted with regularization.

In contrast, if the concepts of the lexicons are understood and easy to inspect, we are also able to easily adapt them by deleting and adding new entries. But more interestingly,

[11] One could also interpret the detection of such generic concepts on a broad range as a featurization of the textual surface into a semantic loosely ordered space for downstream applications. It would allow us to find and model solutions based on these concepts and not on atomic tokens—which are strings or just symbols in the end. Of course the same holds true for applications that try to include more context. In such models these generic concepts could contribute as latent context to the prediction.

[12] For example, the most predictive features (words) of a Logistic Regression classifier or the ones from a NaïveBayes classifier with the most clear-cut priors

we can also combine the concepts. For example, we may build lexicons for different emotional states and use them in one content analysis and then combine them all together for another content analysis where the granularity of emotions is not required but the interest is more on the presence or absence of any description concerning emotions. In a similar way, we may also use different lexicons to compose subsets in a more subtractive way. For example, we could assemble a list of animals but exclude all quadrupeds—just as it is possible with any semantically ordered resource like WordNet. The main restriction for a modeling based on such resources like WordNet is that the available semantic axes in those resources (i.e., the ontological knowledge) must be in line with concepts of the task at hand. If the rigid frame of the given resource is not adequate for the task, the manual effort to curate an apt semantic resource often increases rapidly.

To mitigate this problem, we have proposed a way to derive lexicons which is geared towards facilitating the creation of any kind of lexical-semantic resource. Similarly to the aforementioned combinatory aspect of lexical resources, there are also opportunities to use already available resources to influence or guide the creation of new resources. We addressed such a scenario in Section 6.2.1 and 6.2.2 where we have combined two concepts to create a new, more specific resource.

Another advantage of our approach is the way in which the described workflows are applicable to derive such lexical resources. Those workflows lie in a continuum between purely knowledge-driven application to almost purely data-driven methods. Furthermore, we believe that one of the most important potentials of the approach is the flexibility which such lexical resources provide.

We do not argue categorically against machine learning applications that rely solely (and hence heavily) on annotated data. But we think that reducing the opaqueness of the models is one of the most important prerequisites that we need to fulfill in order to further foster the application of these techniques in the social sciences[13]. Additionally, we emphasize the re-usability of lexical resources for other tasks, which also makes them more sustainable[14].

[13] As we have already mentioned in Chapter 2, the transparency and interpretability of such models (Lipton, 2018, Doshi-Velez and Kim, 2017) is an important element. Fortunately, in recent research, also neural net based methods are subject to closer scrutinizing concerning interpretation of their functionality. See Belinkov and Glass (2019) for an overview of different techniques which have been developed to "open the blackbox".

[14] Of course, such repeated usage has the advantage of comparability. Consequently, the appropriateness of resources which were derived from other labelled data or in completely independent environments (such as LIWC (Pennebaker et al., 2001) or the General Inquirer (Stone et al., 1966)) has to be tested (see also Grimmer and Stewart (2013))

10.5 Including Syntax

One might also encounter the situation when the limiting factors of the presented approach become a noticeable impediment to develop an application. More concretely, the focus of the analysis on what the text is *about*—remember this is what we try to identify with the concept detectors—can potentially introduce problems.

For example, if we want to discern information on the risks and chances of a new technology, one might not be happy with the result if we attribute all sentences to the "risk" class which mention a danger, although this is clearly a part of the concept "risk". A specific danger could also be *banned* using the new technology, leading to the avoidance or prevention of the danger—clearly a sign of a chance.

The reason for this lack of distinction is the simplification of the approach which models sentences simply as sums of filtered bag-of-embeddings. Of course, one could turn to a more sophisticated modeling for the sentence level in the embedding realm as we have described in section 3.1.4. On the other hand, one might strive to preserve the features and properties of the proposed approach and just combine it with minimal syntactic filtering. Hence, we would check for means of language which reverse the content (e.g., negation, diminishing, or putting the content into an irrealis context).

We highly recommend stacking such methods, i.e., combining the strength of different axes of analysis (e.g., coarse semantic modeling, and abstract static syntactic patterns), rather than forcibly integrating one into the other. Such a combination of syntax and semantics is in a way inspired by the tradition of constraint grammar (cf. Karlsson (1990) and Karlsson et al. (1995)). Although the intention of the authors of the framework is rather the linguistic analysis[15], the idea may also be used the other way around. Thus, the inclusion of a linguistic analysis can be beneficial for the prediction on the level of the content analysis.

In a sense, we have already used the static analysis of a linguistic preprocessing component by filtering on part-of-speech tags of words. We opted for such a filtering because of the empirical evidence that concentration of the semantic-bearing parts could benefit the sentence modeling (or as reversed argumentation: that function words pollute such a sentence embedding since the word embedding of the function words cannot link to clear-cut semantics). Nothing prohibits the usage of other grammatical constraints to tackle the problems of the task at hand. Imagine, that for example the focus on the subject of the sentence could deliver better results since it is a good proxy to filter

[15] In constraint grammar, the accomplishment of parsing natural language is enhanced through the integrated application of morphological, syntactical, and semantical rules. Also Linguistic analysis is a formidable field for the application of lexical-semantic resources as we have presented them here.

out disturbing signals—one should not hesitate to apply such additional filtering for information if it leads to the desired result.

10.6 Including (more) Context

The proposed version of measuring the similarity of a sentence embedding against a distributed representation of lexical resources may provide a useful baseline for application. But certainly the inclusion of the document-wide or paragraph-wide context could add some additional layer to sharpen the instrument of analysis.

While the local context of the sentence already leads to a certain disambiguation concerning the lexical ambiguity, the addition of a global or at least broader context than the current sentence could prove beneficial for content analysis. For example, consider the sentence

> *The Reds have ascended the highest throne.*

Given the sentence alone, it is not clear if "The Reds" refers to a communist party that has won the election or if the Liverpool F.C. has won the Champions League. But surely the inclusion of the surrounding sentences (or even a broad abstraction from the whole document) would have provided disambiguation opportunities.

Such (ad hoc) word sense disambiguation on the word embedding level has been proposed by Arora et al. (2018) and is certainly helpful to improve the quality of the analysis. On the other hand, simply taking the surrounding sentences into account (as a weighted dynamic co-importance of concepts), may evenly lead to the desired improvement.

Additionally, we should restrict the influence of those contexts in a meaningful manner—either by signal strength or by additional axes of prior information. Suppose that we found a clear signal in the title or header of an article; this should clearly have more influence than an arbitrary nearby sentence in the text. We conjecture therefore that the inclusion of context in this fashion turns out to be fruitful for many application cases.

While these possibilities must remain a subject of speculation for the moment, we would like to highlight the property of the approach that such enhancements on the modeling level can easily be added and freely combined. We hope that the research community explores the potentially vast field of application at its demand.

11

Conclusion

> "When in doubt tell the truth."
> — Mark Twain

```
In [246]: analogy(a="Untersuchung", b="Ergebnis", x="Fazit", y=None,
model_given=model, verbose=True)
'Untersuchung' is to 'Ergebnis' as 'Fazit' is to 'Resultat'
Out[246]:
[('Resultat', 0.6857106685638428),
 ('Bilanz', 0.5121608376502991),
 ('Schlussfolgerung', 0.5056539177894592),
 ('Grundtenor', 0.4616406559944153),
 ('Zwischenbilanz', 0.4595254957675934),
 ('Befund', 0.45006388425827026),
 ('Endresultat', 0.42085230350494385),
 ('Quintessenz', 0.41845616698265076),
 ('bilanzieren', 0.4164115786552429),
 ('Folgerung', 0.41628509759902954)]
```

In this chapter, we refer to our research questions while summarizing the most important points in this thesis.

To address the first two research questions, we briefly review the challenges that we have identified and the strategy we have applied to counteract them. In this light, we also

refer to the design decisions that we have made for the approach based on the strategy. As a next point, we also take a glimpse on the implementation of the software which we release accompanying this thesis. Before we round up this chapter with a short description of the thesis in a nutshell, the most important results and insights from the empirical part of this work are concisely summarized.

In the first two chapters we have attempted to set the scene for this piece of research. We have intentionally contrasted the promising development of interdisciplinary work between computational linguistics and the social sciences with the pitfalls that other researchers in this domain have already identified. Further, we also discussed on the general requirements of interdisciplinary research.

As one of the main conclusions from this part, we located an area of possible application so that we include the link to an established methodology in one field while retaining the ambition to create a solution that at the same time fulfills requirements of the other field. As a result, we have derived a catalogue of desiderata to respect while designing and implementing the approach. We repeat here a part from the conclusion of Chapter 2:

Besides the general criteria for content analysis as a method (validity and reliability), we especially emphasize transparency which leads us to further rely on simplicity and, more architecturally, on modularity. This in turn allows us to also strive for flexibility and versatility, finally fostering the goal of sustainability.

Finally, it will remain up to the reader if we were successful in meeting this catalogue of desiderata and design principles while implementing a working solution for a given set of problems. However, Chapter 1 and 2 address the first two research questions, aimed at identifying challenges of interdisciplinary research and defining criteria for the case at hand.

With respect to the proposed solution we consider the different steps (lexicon induction, concept detector creation, and classification) as clearly separable. Thus, we achieve a level of modularity that allows us to exchange single components or to make use of single components in a stand-alone fashion.

The different stages are also linked to the guideline of transparency in the sense that inspection of the (intermediate) results is possible at every stage. This is valid for the resource as well as for the application and hence its model for prediction. Since the whole approach may also be applied in a chained fashion—relying mostly on default parametrization—it also works as an end-to-end pipeline with interpretable intermediate results.

In order to preserve these desired properties, the software is implemented according to the conceptual modularity (see Appendix A). While we have not exploited the full potential of the implementation in the empirical part of this work[1]—i.e., we have seldomly engaged in the in-depth analysis and only pointed tentatively to the cross-lingual applicability—we demonstrated that it is applicable for several languages with comparable performance[2].

The experiments on document classification, with a special focus on small and skewed data sets, demonstrate that the approach outperforms a baseline relying on standard methods for text classification. While the overall accuracy is not that different, especially the evaluation on the macro measurements show the effectiveness for the intended usage—which is to represent all concepts (or classes) with a comparable quality. Since the mere quantitative assessment of the classifier and the negligence of the differences in the performance for the single categories may have drastic outcomes for automated content analyses, we focus on this specific challenge. The given approach tackles the problem efficiently through the externalization of generalization by deriving the resources from an independent general word embedding as a model of meaning for a language.

The second extrinsic evaluation scenario concerning framing detection points to evidence that the approach is also suited for more intricate challenges in automated content analysis. In this case, especially the double data skew—resulting from an over-represented residual class and a skew in the distribution over the different classes—caused problems for standard approaches. Again, as in the document classification, the externalized generalization alleviated the problem, although the level of the class-wise performance still differs in the fine-grained framing classification. Furthermore, we could omit the conceptualization of the residual class which cannot be spared in discriminatory classifiers.

The second property that made this task hard to solve is that the relevant information is locally strongly bounded, i.e., the spans of text that triggered the annotation of a frame were often very short. This leads in turn to difficulties stemming from standard BoW modeling for documents (or paragraphs). Because this simplified modeling includes all parts of the given text equally (besides standard filtering and weighting), it is not suited to generalize the pattern for recognition if the true signal is not corresponding with the unit of analysis. Our proposal to tackle this problem benefits from a natural requirement from the application of the lexical resources, namely to apply them on the sentence level. By shifting the level of prediction down to the sentence level and thus narrowing the

[1] This refers also to the fact that we have not investigated on the potential benefit from more structured modeling which is available due to the underlying full-fledged dependency parse.

[2] The centroids reported in Appendix B for German, French, and English are computed in the same way for each language but completely independent. Thus, this case for the three languages does not refer to the cross-lingual application we proposed in Chapter 10. We add these resources to the Appendix B to illustrate that the implementation works for each of the given languages

window of text on which we operate, we enable modes of prediction particularly for such strongly locally bounded information. In the experiments, we demonstrated that we improve on almost all scenarios for the case at hand.

As a last part, we also included a critical discussion and assessment of possible limitations of the approach. The main limitation in our view is that due to its focus on the theme of the content (what the text is *about*), the approach is certainly not apt for all sorts of content analysis. Additionally, we have proposed a number of possible remedies for different limitations of the approach and connected it to other research. Finally, we outlined several alternative ways of application, addressing the last group of research questions which we listed in Chapter 1.

This Work in a Nutshell

This thesis is about the application of techniques from computational linguistics for approaches in social sciences that use the method of content analysis with a clear focus on textual data. Firstly, we address the challenge of interdisciplinary work and identify the most important desiderata and pitfalls to avoid when porting methods to other fields.

Based on guidelines that we derive from these desiderata, we present an approach which connects to the dictionary-based methods that are pervasively present in automated content analysis. In order to perform well, curated lexicons are needed. This thesis presents a way to facilitate the largely automated induction of such lexical resources.

Furthermore, we propose a simple way to apply these lexical resources in an embedded modeling of the textual material. The automated creation of such resources and their application addresses the research questions about the concrete solution for these challenges. It is also an answer to the question about how to accomplish this goal with respect to the desiderata concerning transparency on the level of the model, its components, and its predictions.

In order to intrinsically and extrinsically evaluate the proposed approach, we present an extensive set of experiments, covering lexicon induction, document classification, and framing detection. The measured performance of classification is between satisfying and excellent while outperforming the baselines in a wide range of applications.

As a special property, the approach is, on the one hand, designed to tackle cases where only small sets of annotated data are available that also consist of a heavy skew in the class distribution. On the other hand, the method is also suitable for cases where

the important information in the text is strongly locally bounded (as in the framing detection).

In addition to the given empirical evaluation, we also connect our work to current research and propose some specific improvements to tackle specific challenges and mitigate or overcome identified limitations. Additionally, we have striven to elucidate unexploited potential of the approach and pointed tentatively into the direction of possible applications, such as cross-lingual scenarios.

While the scope of the empirical evaluation had to be narrowed in order to keep the focus, we suggest taking the given evidence as a sign for the valid contribution of the approach. Having laid out a set of options to adapt and develop it further, we leave it up to the research communities to try it out and would be happy to provide them with basic advice for the interested scholars.

A

Implementation details

In this part, we briefly explain the core components of the software that we release together with the publication of this thesis. Note that this section contains rather high-level descriptions of the components as the specific implementation may change in future to improve performance or quality.

A.1 SeedFinder

The SeedFinder class is a basic helper to identify the most important words of a category given a set of instances of labelled data. We have referred to this set of words occasionally as "core of the concept". While it it perfectly fine to define such a set purely knowledge-driven, the SeedFinder offers the inclusion of a data-driven method and should be viewed as a complementary method.

Since it includes a mixture of a generic frequency comparison (which could be thought of as a global version TF-IDF) and a comparison between the given categories, it yields the best results, if annotated data (from all categories) is available. However, if there is only one category, only the global part of the criteria will apply, hence yielding the words that occur unusually often in the single class.

A.2 LexExpander

This component is for the extension of a given core of a lexicon (which in turn is a collection of terms related to a concept of interest). In its basic configuration (see also Chapter 4), it takes a starting point (where the search should start) and performs a searches for candidates to extend the lexicon. Based on the parametrization (especially the lexicon weights), the assessment will be more daring (including more new terms based on less evidence) or cautious. For extension of a small core, one should start rather daring and get more cautious. For the case, where one wants to fill the gaps of an already existing lexicon, one should choose a rather cautious parametrization.

However, it cannot be safely predicted for each scenario which parametrization will serve the purpose the best. As a rule of thumb, as we have already mentioned in the thesis, one should try (slightly) different starting points rather than searching in the same direction for a large number of iterations. Although the increase of random (re-assemble the following starting points farther away from the current search point and the lexicon) in the search process may render long searches productive, this setting tends to be more surprising and maybe lead into unwanted directions.

A.3 LexEmbedder

This Component takes basically a lexicon, gets the vector representations of the terms and then clusters the lexicon according to the given clustering parametrization. Additionally, the result is shown as well as a brief evaluation on how many words of the lexicon are close to the calculated centroids. This allows on the one hand to manually inspect the concepts detectors (centroids), and, on the other hand, to estimate the coverage of the lexicon in the centroid (which terms are represented and how well).

This component allows to re-use the same embedding model as the LexExpander (or any of the other components relying on an embedding), hence allowing for parallel or chained usage without multiple instances of the embedding loaded into RAM (since embedding models tend to be quite large).

A.4 SentEmbedder

Since we perform the comparison (and hence the classification) with the concept detectors on the sentence-level, this class allows to easily add this layer of representation on the sentence object (see the description of the conll_reader_utils below for the sentence

objects). More precisely, we just add additional attributes which contain a list of the embedded words, the sentence representation as a vector, and an array of similarity scores to the centroids.

Although the current implementation is traversing iteratively over the sentences, it should be obvious that the sentence objects contain after this transformation the information for the classification independently. Hence, the parallelization of the process (and further downstream processing) is easily applicable. For a limited number of classes (represented each by n concept detectors) the comparison of the sentence to the concept detectors is not a bottleneck. However, if one wants to simultaneously compare the sentence to thousand of concepts (maybe represented by ten thousands of concept detectors)—this is a case for universal information extraction—one maybe needs to encapsulate this step which is then easily doable due to the data-centric modeling.

A.5 UClassifier

This is the implementation of the (unsupervised; hence the U) heuristic classifier that we applied in the experiments. It is basically suited for any classification and can be easily adapted by replacing the predictor component. In its current form, there are only two main parameters that can be set, namely the threshold (a cut-off for similarities below the threshold) and the n-best parameter (also a cut-off to restrict the similarities taken into account for the prediction). In standard cases, these two parameters allow the user to set what level of certainty he wants to include into the overall prediction. Additionally, the number of n-best should be adapted when the number of classes is large (see also Chapter 5).

Note that the classifier produces a label (including an estimation for its probability) for the classes for each sentence, turning finally these evaluations into an aggregation over the piece of text that is fed in. If multi-label classification is desired, one would just adjust the aggregation of the sentence-wise or document-wise output so that we do not filter for the highest scoring class.

A.6 Utilities

A.6.1 conll_reader_utils

This utility class reads in CONLL-based format which is a de-facto standard for the output of a parser. It is a shallow converter that renders CONLL string output into sentence objects (consisting also of nested token objects) which are subsequently the basic layer of abstraction, i.e., the parse of a text in CONLL-format output gets converted into a list of sentence objects.

Furthermore, it should allow us to write much more readable and concise code, in the sense that we have modelled the tokens with attributes that represent the columns in the CONLL-format

A.6.2 CachedEmbedding

Since we use the embedding extensively for look-ups (to get the vector representation for words and word combinations), we have written this small wrapper class which consists of a (configurable) lru-cache (last recent unit cache) provided by python language[1]. In this way the sentence representation (based on word embedding look-ups) is created noticeably quicker, especially if we have larger data-sets (hundreds or thousands of texts). It also speeds up the LexExpander remarkably.

A.6.3 reporting_tools

This contains some tools for simplified access to standard evaluations, including the creation of according confusion matrices[2]. Additionally, we have also included some tools to visualize the outcome of the re-embedding process of the lexicons that produces the concept detectors. This allows us also to inspect their relative position to each other in the embedded space, as well as to any given word.

[1] See `https://docs.python.org/3/library/functools.html`

[2] Since we have created this utility the standard evaluation of `sklearn` has also become more verbose and is now also reporting macro scores.

B

Lexical Resources

B.1 Lexical Resources for the Document Classification Task

In this section of the Appendix, we illustrate all the lexical resources that were derived for the document classification task. For the sake of clarity and consistency, we choose the form of the re-embedded lexicons, represented by the centroids of the quantized lexicons, i.e., displaying the vicinities of those.

B.1.1 Resources for the Domain *Bildung (Education)*

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Studiengang	0.8779
2	Studierende	0.8237
3	Fachhochschule	0.8209
4	Bachelor-	0.8144
5	Bachelor	0.8125
6	Masterstudiengang	0.7992
7	Masterstudium	0.7926
8	Studienrichtung	0.7894
9	Absolvent	0.7867
10	Masterstufe	0.7813

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Volksschule	0.7286
2	Bildungsdirektion	0.7120
3	Erziehungsdirektorenkonferenz	0.7012
4	Mittelschule	0.6788
5	Lehrkraft	0.6775
6	Bildungsrat	0.6684
7	Lehrperson	0.6516
8	Lehrplan	0.6510
9	Mindestpensen	0.6501
10	Lehrpersonenkonferenz	0.6476

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	pädagogisch_Hochschule	0.8392
2	Fachhochschule	0.8011
3	Studiengang	0.7796
4	Angewandte_Linguistik	0.7548
5	Berufsschule	0.7480
6	hoch_Fachschule	0.7461
7	Mittelschule	0.7461
8	Lehrgang	0.7454
9	PHZH	0.7432
10	PH_Zürich	0.7395

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Oberstufe	0.8695
2	Primarschule	0.8528
3	Sekundarschule	0.8339
4	Sekundarstufe	0.8273
5	Volksschule	0.8210
6	Lehrperson	0.8205
7	Lehrkraft	0.8133
8	Unterstufe	0.8005
9	Mittelstufe	0.7991
10	Grundstufe	0.7977

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Universität	0.8003
2	Student	0.7982
3	Uni	0.7872
4	Dozent	0.7650
5	Professor	0.7441
6	Doktorand	0.7437
7	Studierende	0.7212
8	Harvard	0.7167
9	Studentin	0.7129
10	Eliteuniversität	0.6997

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Gymnasium	0.8701
2	Mittelschule	0.8493
3	Langgymnasium	0.8275
4	Langzeitgymnasium	0.8039
5	Kurzzeitgymnasium	0.8005
6	Sekundarschule	0.8003
7	Aufnahmeprüfung	0.7918
8	Kurzgymnasium	0.7898
9	Sekundarstufe	0.7661
10	Gymi	0.7607

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Schüler	0.8346
2	Primarschüler	0.8345
3	Kindergarten	0.8235
4	Unterricht	0.8227
5	Oberstufe	0.8213
6	Primarschule	0.8176
7	Mittelstufe	0.8118
8	Sekundarschule	0.7924
9	Schule	0.7920
10	Klassenzimmer	0.7884

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Zürcher_Hochschule	0.8525
2	Hochschule	0.8378
3	angewandte_Wissenschaft	0.8019
4	Wädenswil_ZHAW	0.7986
5	Kunst_ZHdK	0.7961
6	Kunst_HGKZ	0.7926
7	ZHAW	0.7922
8	Kunst_ZHDK	0.7822
9	ZHdK	0.7723
10	Angewandte_Wissenschaft	0.7634

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Berufsmaturität	0.7952
2	Fachmittelschule	0.7943
3	Gymnasium	0.7875
4	Mittelschule	0.7870
5	Berufslehre	0.7717
6	Matur	0.7716
7	Fachmaturität	0.7711
8	Maturität	0.7598
9	Berufsmatur	0.7558
10	Berufsmittelschule	0.7544

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Uni	0.8452
2	Fakultät	0.8173
3	ETH	0.8005
4	medizinisch_Fakultät	0.7896
5	Uni_Zürich	0.7855
6	Universität	0.7698
7	Universität_Zürich	0.7678
8	Fachhochschule	0.7678
9	Studierende	0.7580
10	Doktorand	0.7549

TABLE B.1: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for *Bildung/Schule/Hochschule (Education/School/University)* in the semantic space of the word2vec model, ordered by cosine similarity

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Forschungsergebnis	0.8064
2	wissenschaftlich	0.7992
3	wissenschaftlich_Erkenntnis	0.7466
4	empirisch	0.7421
5	Forschungsarbeit	0.7247
6	Forschungsresultat	0.7092
7	Forschungserkenntnis	0.7048
8	neurowissenschaftlich	0.6964
9	Forschungsansatz	0.6902
10	Hirnforschung	0.6895

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Hirnforschung	0.7917
2	Biologie	0.7884
3	Neurowissenschaft	0.7649
4	Anthropologie	0.7609
5	Wissenschaft	0.7575
6	Neurobiologie	0.7471
7	Naturwissenschaft	0.7262
8	naturwissenschaftlich	0.7241
9	Psychologie	0.7147
10	Geisteswissenschaft	0.7136

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Zellbiologie	0.8018
2	Biochemie	0.7901
3	Mikrobiologie	0.7861
4	Verhaltensbiologie	0.7769
5	Pflanzenbiologie	0.7727
6	Molekularbiologie	0.7686
7	Universität_Bern	0.7638
8	theoretisch_Physik	0.7566
9	Umweltphysik	0.7515
10	Pflanzenwissenschaft	0.7478

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Soziologie	0.8664
2	Rechtswissenschaft	0.7979
3	Germanistik	0.7972
4	Wirtschaftswissenschaft	0.7935
5	Psychologie	0.7914
6	Literaturwissenschaft	0.7896
7	Ethnologie	0.7833
8	Kulturwissenschaft	0.7819
9	Islamwissenschaft	0.7786
10	Religionswissenschaft	0.7782

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Biotechnologie	0.8015
2	Materialwissenschaft	0.7978
3	Life_Sciences	0.7393
4	Life-Sciences	0.7333
5	Biowissenschaften	0.7333
6	Life-Science	0.7322
7	Verfahrenstechnik	0.7311
8	Pharmazie	0.7291
9	Informatik	0.7198
10	Elektrotechnik	0.7176

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Technologie	0.8249
2	technologisch	0.7523
3	Innovation	0.7359
4	technisch_Innovation	0.7281
5	Informationstechnologie	0.7135
6	technologisch_Fortschritt	0.7077
7	technologisch_Entwicklung	0.7043
8	modern_Technologie	0.6893
9	technisch_Fortschritt	0.6781
10	Technik	0.6781

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Grundlagenforschung	0.7833
2	Forschung	0.7435
3	Forschungsgebiet	0.7315
4	regenerativ_Medizin	0.7235
5	Forschungsbereich	0.7171
6	Forschungsschwerpunkt	0.7054
7	forschen	0.7050
8	Systembiologie	0.7022
9	Biosysteme	0.7017
10	Neurowissenschaft	0.7010

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Grundlagenforschung	0.8369
2	Forschung	0.8165
3	angewandte_Forschung	0.7449
4	Nationalfond	0.7271
5	anwendungsorientiert_Forschung	0.7182
6	anwendungsorientiert	0.7181
7	Spitzenforschung	0.7012
8	Forschungsprojekt	0.6996
9	Forschungsbereich	0.6970
10	universitär	0.6961

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Mathematik	0.8522
2	Naturwissenschaft	0.8220
3	Chemie_Physik	0.7768
4	Mathematik_Naturwissenschaft	0.7562
5	Biologie_Chemie	0.7555
6	Mathematik_Deutsch	0.7550
7	Physik	0.7484
8	Biologie	0.7403
9	Fach_Deutsch	0.7403
10	Physik_Chemie	0.7374

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	physikalisch	0.7593
2	Quantentheorie	0.7445
3	Quantenmechanik	0.7444
4	menschlich_Gehirn	0.7392
5	molekular	0.7237
6	Hochtemperatursupraleitung	0.7221
7	Quantenphysik	0.7161
8	biologisch_Evolution	0.7075
9	Supraleitung	0.7046
10	Elektromagnetismus	0.7044

TABLE B.2: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for *Wissenschaft/Forschung/Technologie (Science/Research/Technology)* in the semantic space of the word2vec model, ordered by cosine similarity

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Studium	0.8711
2	Lizenziat	0.8150
3	Doktorat	0.8102
4	Wirtschaftsstudium	0.7970
5	Lizentiat	0.7692
6	Geschichtsstudium	0.7654
7	Zusatzstudium	0.7609
8	Jusstudium	0.7606
9	Matura	0.7599
10	Zweitstudium	0.7582

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Ausbildung	0.8670
2	Grundausbildung	0.8214
3	Lehrgang	0.7995
4	Zusatzausbildung	0.7838
5	ausbilden	0.7410
6	einjährig_Ausbildung	0.7401
7	zweijährig_Ausbildung	0.7398
8	ausgebildet	0.7389
9	weiterbilden	0.7387
10	berufsbegleitend	0.7381

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Matura	0.8211
2	kaufmännisch_Lehre	0.8193
3	Handelsschule	0.8045
4	KV-Lehre	0.8016
5	Handelsdiplom	0.7850
6	kaufmännisch_Ausbildung	0.7782
7	Praktikum	0.7762
8	Matur	0.7650
9	Berufsmatura	0.7455
10	Schreinerlehre	0.7452

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	vierjährig_Lehre	0.7720
2	dreijährig_Lehre	0.7696
3	Lehrling	0.7461
4	Polygrafen	0.7315
5	Verkaufslehre	0.7286
6	Schnupperlehre	0.7246
7	Praktikum	0.7245
8	Ausbildung	0.7202
9	Lehrabschlussprüfung	0.7199
10	viert_Lehrjahr	0.7137

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Lehrstelle	0.8553
2	Ausbildungsplatz	0.8083
3	Brückenangebot	0.8018
4	Schulabgänger	0.7990
5	Lehrbetrieb	0.7899
6	Lernende	0.7803
7	10_Schuljahr	0.7556
8	Praktikumsplatz	0.7481
9	Schulabgängerin	0.7462
10	Auszubildende	0.7446

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Automechaniker	0.8096
2	Elektromonteur	0.8088
3	Schreiner	0.7887
4	Polymechaniker	0.7741
5	Hochbauzeichner	0.7488
6	Automatiker	0.7323
7	Betriebspraktiker	0.7296
8	Bauzeichner	0.7277
9	Sanitärmonteur	0.7216
10	Landschaftsgärtner	0.7206

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Berufslehre	0.8280
2	Berufsmaturität	0.8063
3	Berufsausbildung	0.8041
4	Berufsmatura	0.8020
5	Berufsmatur	0.8017
6	Fachhochschulstudium	0.7891
7	berufsbegleitend	0.7835
8	Matur	0.7804
9	Ausbildung	0.7789
10	Matura	0.7760

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Weiterbildung	0.8249
2	Weiterbildungsangebot	0.8080
3	Ausbildungsgang	0.7920
4	Lehrgang	0.7670
5	hoch_Fachschule	0.7439
6	Weiterbildungsmöglichkeit	0.7365
7	Ausbildungsangebot	0.7353
8	berufsbegleitend	0.7332
9	Studiengang	0.7315
10	Bildungsgang	0.7296

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	kaufmännisch_Angestellte	0.8076
2	Kauffrau	0.7997
3	diplomierte	0.7885
4	Coiffeuse	0.7645
5	Fachfrau_Betreuung	0.7475
6	Floristin	0.7361
7	gelernt	0.7240
8	Autolackiererin	0.7180
9	Hochbauzeichnerin	0.7176
10	Primarlehrerin	0.7069

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Beruf	0.7895
2	Berufsfeld	0.7621
3	Lehrerberuf	0.7606
4	Lehrberuf	0.7602
5	handwerklich_Beruf	0.7170
6	Berufsleute	0.7158
7	Berufswelt	0.7104
8	Berufslehre	0.7102
9	kaufmännisch_Bereich	0.7072
10	Berufsleben	0.7014

TABLE B.3: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for *Beruf/Berufsbildung (Professions/Vocational training)* in the semantic space of the word2vec model, ordered by cosine similarity

B.1.2 Resources for the Domain *Umwelt (Environment)*

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Küchenabfall	0.8232
2	Grüngut	0.8081
3	Grünabfall	0.7981
4	vergärt	0.7974
5	Klärschlamm	0.7879
6	Bioabfall	0.7857
7	organisch_Abfall	0.7802
8	Bioabfälle	0.7703
9	Grüngutabfall	0.7655
10	Kompost	0.7643

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Kläranlage	0.8788
2	Abwasser	0.8679
3	ARA	0.8161
4	Abwasserreinigungsanlage	0.8050
5	Abwasserreinigungsanlage_ARA	0.7675
6	gereinigt_Abwasser	0.7592
7	Fremdwasser	0.7535
8	Schmutzwasser	0.7527
9	Abwasserreinigung	0.7357
10	Grundwasser	0.7307

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Sammelstelle	0.7988
2	mobil_Sammelstelle	0.7915
3	Kehricht	0.7685
4	Abfall	0.7647
5	entsorgen	0.7536
6	Grüngut	0.7413
7	illegal_deponiert	0.7268
8	Kleinmetall	0.7164
9	Betriebskehricht	0.7151
10	Hauskehricht	0.7112

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	ERZ	0.8763
2	Recycling_Zürich	0.8612
3	Entsorgung	0.8138
4	Leta_Filli	0.8088
5	Recycling_ERZ	0.8022
6	Entsorgung_+	0.7770
7	ERZ_Entsorgung	0.7324
8	&_Recycling	0.7079
9	Recycling	0.6472
10	Abfall	0.6206

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	entsorgen	0.8760
2	Abfall	0.8629
3	Müll	0.8328
4	Kehricht	0.8170
5	Abfallsack	0.8015
6	Kehrichtsack	0.7778
7	Sperrgut	0.7685
8	Hauskehricht	0.7671
9	öffentlich_Abfalleimer	0.7667
10	Altpapier	0.7542

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Fernwärmenetz	0.7910
2	Hagenholz	0.7876
3	KVA	0.7752
4	Kehrichtverbrennungsanlage_Josefstrasse	0.7719
5	Holzheizkraftwerk	0.7669
6	Fernwärme	0.7551
7	Kehrichtverbrennungsanlage	0.7540
8	Klärwerk_Werdhölzli	0.7366
9	Werk_Hagenholz	0.7351
10	Kehrichtheizkraftwerk_Hagenholz	0.7317

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Recycling	0.8041
2	Elektroschrott	0.7766
3	Altglas	0.7733
4	wiederverwertet	0.7719
5	Wiederverwertung	0.7691
6	Abfall	0.7551
7	rezyklieren	0.7537
8	entsorgen	0.7343
9	Sonderabfall	0.7238
10	Wertstoff	0.7117

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Reststoffdeponie	0.7601
2	Kompogasanlage	0.7548
3	Kompogas-Anlage	0.7480
4	Kehrichtverwertungsanlage	0.7415
5	Kezo	0.7332
6	Reaktordeponie	0.7309
7	Kehrichtverbrennungsanlage	0.7196
8	Deponie	0.7101
9	KVA	0.7076
10	Kompogas_AG	0.6957

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Kehricht	0.7842
2	Abfall	0.7797
3	Reststoff	0.7694
4	Bauabfall	0.7687
5	Schlacke	0.7657
6	Kehrichtverbrennungsanlage	0.7648
7	Haushaltabfall	0.7619
8	Siedlungsabfall	0.7583
9	Klärschlamm	0.7516
10	Kehrichtschlacke	0.7514

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Mülldeponie	0.7975
2	Deponie	0.7865
3	Abfalldeponie	0.7628
4	Müll	0.7622
5	Abfall	0.7448
6	illegal_Deponie	0.7423
7	Abfallberg	0.7307
8	Kehrichtdeponie	0.7280
9	Verbrennungsanlage	0.7213
10	giftig_Abfall	0.7123

TABLE B.4: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for *Abfall (Waste)* in the semantic space of the word2vec model, ordered by cosine similarity

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Feuchtigkeit	0.7518
2	Sonnenlicht	0.7514
3	Sonneneinstrahlung	0.7442
4	Wasserdampf	0.7351
5	Wassertröpfchen	0.7149
6	Luftschicht	0.7004
7	verdampfend	0.6981
8	Lufthülle	0.6849
9	Luftdruck	0.6801
10	Oberfläche	0.6777

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Klimamodell	0.8256
2	Massenbilanz	0.7944
3	Computermodell	0.7767
4	regional_Klimamodell	0.7653
5	Klimaentwicklung	0.7619
6	Klimasimulation	0.7514
7	Satellitendaten	0.7479
8	Klimadaten	0.7424
9	Messdaten	0.7417
10	Temperaturverlauf	0.7315

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	IPCC	0.8442
2	Weltklimarat_IPCC	0.8091
3	Weltklimarat	0.7754
4	Klimaforscher	0.7689
5	Uno-Klimarat_IPCC	0.7672
6	Klimabericht	0.7625
7	IPCC-Bericht	0.7579
8	Uno-Weltklimarat_IPCC	0.7541
9	Uno-Klimarat	0.7448
10	UNO-Weltklimarat_IPCC	0.7365

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	frieren	0.6898
2	Badetuch	0.6784
3	schwitzen	0.6540
4	Liegestuhl	0.6310
5	wärmen	0.6269
6	baden	0.6262
7	nass	0.6183
8	Badehose	0.6110
9	Glace_schlecken	0.6046
10	Tuch	0.6012

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Gewitter	0.7815
2	Schneefall	0.7710
3	Kaltfront	0.7396
4	Meteorologen	0.7289
5	Flachland	0.7274
6	Warmfront	0.7260
7	Westwind	0.7244
8	Hochdrucklage	0.7230
9	arktisch_Kaltluft	0.7199
10	Niederschlag	0.7166

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Klimawandel	0.8982
2	Klimaerwärmung	0.8981
3	Erderwärmung	0.8699
4	Klimaveränderung	0.8682
5	global_Erwärmung	0.8353
6	Treibhauseffekt	0.7857
7	Erwärmung	0.7798
8	Klimaänderung	0.7730
9	global_Klimaerwärmung	0.7432
10	Mensch_verursacht	0.7238

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Erwärmung	0.8502
2	Ozonabbau	0.7806
3	Meereis	0.7781
4	Strahlungsbilanz	0.7730
5	Nordhemisphäre	0.7699
6	Temperaturanstieg	0.7693
7	arktisch_Meereis	0.7640
8	CO2-Anstieg	0.7611
9	Eisbedeckung	0.7588
10	Kohlenstoffkreislauf	0.7580

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	ansagen	0.5472
2	ausbleiben	0.5428
3	toben	0.5321
4	anhaltend	0.5285
5	wars	0.5050
6	andauern	0.5034
7	prophezeien	0.5021
8	heftig	0.5008
9	erwartet	0.4964
10	anhalten	0.4936

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	verändert	0.6224
2	Veränderung	0.6191
3	Anpassung	0.6094
4	anpassen	0.5922
5	vernünftig	0.5611
6	optimieren	0.5565
7	angepasst	0.5533
8	verbessert	0.5530
9	Struktur	0.5502
10	Verbesserung	0.5460

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Steiner	0.6191
2	Schuler	0.6151
3	Bühler	0.6118
4	Arnold	0.6068
5	Horat	0.6066
6	Wirz	0.6033
7	Wick	0.5956
8	Spörri	0.5902
9	Bieri	0.5865
10	Wetterschmöcker	0.5858

TABLE B.5: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for *Klima (Climate)* in the semantic space of the word2vec model, ordered by cosine similarity

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Botanik	0.7398
2	antik	0.6802
3	Antike	0.6046
4	phantastisch	0.5982
5	Forstwirtschaft	0.5926
6	Altertum	0.5906
7	Kulturgeschichte	0.5788
8	prähistorisch	0.5736
9	mittelalterlich	0.5672
10	Archäologie	0.5642

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Artenvielfalt	0.8664
2	Lebensraum	0.8483
3	Biodiversität	0.7873
4	Tier-	0.7830
5	Pflanzenart	0.7819
6	artenreich	0.7797
7	vielfältig_Lebensraum	0.7701
8	Pflanzenwelt	0.7550
9	Tierart	0.7503
10	Ökosystem	0.7477

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Naherholungsgebiet	0.8289
2	Erholungsraum	0.8151
3	Erholungsgebiet	0.7916
4	Naturraum	0.7648
5	Kulturlandschaft	0.7638
6	Grünfläche	0.7614
7	naturnah	0.7564
8	Naturschutzgebiet	0.7547
9	Naherholungsraum	0.7516
10	Grünzone	0.7430

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Magerwiese	0.8156
2	Feuchtwiese	0.7830
3	Buntbrache	0.7768
4	Trockenstandort	0.7719
5	Naturwiese	0.7706
6	Riedwiese	0.7660
7	Naturschutzgebiet	0.7586
8	Hochstamm-Obstgarten	0.7584
9	Lebensraum	0.7557
10	Feldgehölz	0.7516

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Weiher	0.8199
2	Altlauf	0.7915
3	Naturschutzgebiet	0.7900
4	Flachufer	0.7786
5	Auenlandschaft	0.7740
6	Flusslauf	0.7676
7	Kiesbank	0.7598
8	renaturieren	0.7559
9	Fluss	0.7551
10	Fliessgewässer	0.7547

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Moorlandschaft	0.8118
2	Landschaftsschutzgebiet	0.8068
3	Flachmoor	0.8022
4	BLN	0.7895
5	Bundesinventar	0.7872
6	national_Bedeutung	0.7810
7	Amphibienlaichgebiet	0.7784
8	geschützt_Landschaft	0.7555
9	Schutzgebiet	0.7547
10	Naturschutzgebiet	0.7437

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Massenverlust	0.7008
2	Messperiode	0.6503
3	Sonnenfleck	0.6460
4	Eisverlust	0.6452
5	Erdatmosphäre	0.6344
6	Sonnenaktivität	0.6341
7	Eisbedeckung	0.6322
8	Erwärmung	0.6286
9	Eiskappe	0.6269
10	Grönlandeis	0.6265

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Pflanze	0.7768
2	Laubbaum	0.7612
3	Baum	0.7497
4	abgestorben	0.7441
5	Same	0.7340
6	Strauch	0.7278
7	Obstbaum	0.7221
8	Laub	0.7150
9	Nadelbaum	0.7118
10	Flechte	0.7111

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	abgelegen	0.5828
2	Moor-	0.5610
3	Tal	0.5577
4	abgeschieden	0.5414
5	entlegen	0.5266
6	Peripherie	0.5254
7	nördlich	0.5240
8	Weiler	0.5229
9	gelegen	0.5194
10	Dorf	0.5173

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Kulturland	0.8476
2	Fruchtfolgefläche	0.8291
3	Landwirtschaftsland	0.8173
4	sogenannt_Fruchtfolgefläche	0.8121
5	Landwirtschaftsfläche	0.7958
6	Siedlungsgebiet	0.7921
7	Landwirtschaftsgebiet	0.7858
8	ökologisch_wertvoll	0.7693
9	ökologisch_Ausgleichsfläche	0.7656
10	Landwirtschaftszone	0.7514

TABLE B.6: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for *Natur/Landschaft (Nature/Landscape)* in the semantic space of the word2vec model, ordered by cosine similarity

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Grundstück	0.9089
2	Parzelle	0.8840
3	Landstück	0.8283
4	Areal	0.8120
5	Überbauung	0.8024
6	Bauland	0.7821
7	5800_Quadratmeter	0.7719
8	Gemeindeland	0.7702
9	gemeindeeigen_Grundstück	0.7628
10	Gemeindegrundstück	0.7546

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	eingezont	0.8520
2	überbauen	0.8518
3	Bauland	0.8418
4	Bauzone	0.7959
5	Baulandreserve	0.7931
6	einzuzonen	0.7875
7	einzonen	0.7854
8	überbaubar	0.7817
9	eingezonten	0.7791
10	Parzelle	0.7744

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Raumplanung	0.8432
2	Richtplan	0.8159
3	raumplanerisch	0.7877
4	Richtplanung	0.7836
5	kantonalen_Richtplan	0.7811
6	Siedlungsentwicklung	0.7509
7	kantonalen_Richtplanung	0.7327
8	Waldentwicklungsplan	0.7271
9	Raumordnungskonzept	0.7190
10	Raumplanungsrecht	0.7140

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	verdichtet_Bau	0.8100
2	Verdichtung	0.8034
3	baulich_Verdichtung	0.8003
4	Siedlungsentwicklung	0.7615
5	Siedlungsqualität	0.7443
6	inner_Verdichtung	0.7353
7	bereits_überbaut	0.7209
8	baulich_Entwicklung	0.7169
9	Grünraum	0.7145
10	bestehend_Siedlungsgebiet	0.7127

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Kulturland	0.8496
2	Landwirtschaftsfläche	0.8131
3	Landwirtschaftsland	0.8051
4	Fruchtfolgefläche	0.7956
5	Landwirtschaftsgebiet	0.7946
6	Siedlungsgebiet	0.7879
7	Naturschutzgebiet	0.7834
8	sogenannt_Fruchtfolgefläche	0.7832
9	ökologisch_wertvoll	0.7754
10	Erholungsgebiet	0.7593

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Gestaltungsplan	0.9112
2	privat_Gestaltungsplan	0.8603
3	Umzonung	0.8254
4	Zonenplan	0.8228
5	Bauprojekt	0.8140
6	Sonderbauvorschrift	0.8089
7	BZO	0.7936
8	Bauvorhaben	0.7898
9	Bauordnung	0.7808
10	Zonenänderung	0.7801

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Zersiedelung	0.8637
2	Zersiedlung	0.8216
3	Siedlungsdruck	0.7425
4	Verstädterung	0.7403
5	Siedlungsfläche	0.7232
6	fortschreitend_Zersiedelung	0.7226
7	Bodenverbrauch	0.7116
8	Siedlungswachstum	0.7083
9	Zubetonierung	0.7044
10	Bevölkerungswachstum	0.7029

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Überbauung	0.9054
2	Wohnüberbauung	0.8414
3	Grossüberbauung	0.8169
4	Wohnbau	0.8016
5	Areal	0.7936
6	Wohnsiedlung	0.7878
7	Neubau	0.7800
8	geplant_Überbauung	0.7760
9	Neuüberbauung	0.7644
10	Gewerbenutzung	0.7618

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Bevölkerungswachstum	0.8712
2	Bevölkerungszuwachs	0.8067
3	Bevölkerungszunahme	0.7786
4	Bevölkerungszahl	0.7584
5	Bevölkerungsentwicklung	0.7536
6	Wohnbautätigkeit	0.7471
7	Einwohnerzahl	0.7260
8	Bautätigkeit	0.7213
9	Bevölkerungsanstieg	0.7042
10	Wohnbevölkerung	0.7030

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Wohnzone	0.8566
2	Landwirtschaftszone	0.8409
3	Erholungszone	0.8252
4	Reservezone	0.8225
5	Kernzone	0.8221
6	Freihaltezone	0.8206
7	Gewerbezone	0.8171
8	Bauzone	0.8117
9	umgezont	0.8058
10	Freihaltezone_umteilen	0.7996

TABLE B.7: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for *Raumplanung (Spatial Planning)* in the semantic space of the word2vec model, ordered by cosine similarity

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Wildhüter	0.8222
2	Jagdinspektor	0.7480
3	M13	0.7365
4	Grossraubtier	0.7351
5	Georg_Brosi	0.7346
6	Wildtier	0.7243
7	Reinhard_Schnidrig	0.7237
8	Raubtier	0.7232
9	Luch	0.7216
10	Hannes_Jenny	0.7206

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Hund	0.8668
2	Katze	0.8595
3	Büsi	0.8084
4	Hündin	0.8056
5	Haustier	0.7988
6	Meerschweinchen	0.7955
7	Vierbeiner	0.7788
8	Schäferhund	0.7756
9	Tier	0.7504
10	Hauskatze	0.7491

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Insekt	0.8826
2	Ameise	0.8029
3	Schnecke	0.8015
4	Tierchen	0.7828
5	Raup	0.7790
6	Wurm	0.7660
7	Käfer	0.7650
8	Spinne	0.7634
9	Fledermaus	0.7625
10	Wespe	0.7625

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Jungtier	0.8363
2	Gehege	0.7989
3	Zürcher_Zoo	0.7942
4	Elefant	0.7865
5	Zoo	0.7838
6	Zoo_Zürich	0.7762
7	Tierpfleger	0.7441
8	Elefantenkuh	0.7403
9	Dickhäuter	0.7376
10	Tier	0.7364

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Braunbär	0.8491
2	M13	0.8157
3	JJ3	0.8070
4	MJ4	0.7756
5	Schaf_reißen	0.7668
6	Bär_JJ3	0.7527
7	Graubund_eingewandert	0.7508
8	JJ2	0.7501
9	JJ1	0.7494
10	Jungbär	0.7476

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Schwein	0.8683
2	Huhn	0.8680
3	Ziege	0.8598
4	Kuh	0.8523
5	Rind	0.8492
6	Schaf	0.8329
7	Kalb	0.8224
8	Schaf_Ziege	0.8056
9	Ziege_Schaf	0.8023
10	Tier	0.7973

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Elefant	0.8406
2	Affe	0.8007
3	Schildkröte	0.7960
4	Krokodil	0.7758
5	Raubkatze	0.7641
6	Giraffe	0.7624
7	Nashorn	0.7582
8	Antilope	0.7476
9	Tier	0.7462
10	Flusspferd	0.7456

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Luchs	0.8252
2	Luch	0.7846
3	Wildtier	0.7709
4	Bartgeier	0.7656
5	Braunbär	0.7650
6	Raubtier	0.7630
7	Rothirsch	0.7489
8	Wildschwein	0.7481
9	wild_lebend	0.7475
10	frei_Wildbahn	0.7409

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Säugetier	0.8366
2	Tier	0.8273
3	Insekt	0.8125
4	Reptil	0.8052
5	Tierart	0.7970
6	Artgenosse	0.7957
7	Nagetier	0.7762
8	Jungtier	0.7607
9	Männchen	0.7601
10	Säuger	0.7567

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Reh	0.8242
2	Raubtier	0.8223
3	Wildschwein	0.8196
4	Luch	0.8096
5	Wildtier	0.7925
6	Wildhüter	0.7902
7	Jäger	0.7874
8	Wolf	0.7787
9	Gämse	0.7717
10	erlegen	0.7698

TABLE B.8: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for *Tiere (Animals)* in the semantic space of the word2vec model, ordered by cosine similarity

B.1.3 Resources for the Domain *Verkehr (Traffic)*

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	alpenquerend_Güterverkehr	0.8174
2	Transitgüterverkehr	0.8144
3	Schiene_verlagern	0.8078
4	Verkehrsverlagerung	0.7786
5	Güterverkehr	0.7784
6	Verlagerungspolitik	0.7765
7	Gütertransit	0.7736
8	Gütertransitverkehr	0.7693
9	alpenquerend_Transitverkehr	0.7617
10	Transitschwerverkehr	0.7343

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Güterverkehr	0.8725
2	kombiniert_Verkehr	0.8260
3	Gütertransport	0.8076
4	Schienengüterverkehr	0.7752
5	Kombiverkehr	0.7709
6	Binnenverkehr	0.7679
7	Hupac	0.7644
8	Schienenverkehr	0.7618
9	Personenverkehr	0.7608
10	Wagenladungsverkehr	0.7537

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Seefracht	0.8782
2	Luftfracht	0.8648
3	transportiert_Volumen	0.8348
4	befördert_Tonnage	0.8252
5	Kontraktlogistik	0.7802
6	Landverkehr	0.6568
7	währungsbereinigt	0.6359
8	Panalpina	0.6338
9	Volumenzuwachs	0.6284
10	Transportvolumen	0.6260

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Logistikkonzern_Kühne	0.8400
2	+_Nagel	0.8149
3	&_Nagel	0.8049
4	Panalpina	0.7994
5	Kühne_+	0.7954
6	Kühne_&	0.7837
7	Logistikkonzern	0.7430
8	Klaus-Michael_Kühne	0.6933
9	Logistikkonzern_Panalpina	0.6750
10	Peter_Ulber	0.6346

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	SBB_Cargo	0.8725
2	Gütertochter	0.8150
3	Güterbahn	0.8148
4	BLS_Cargo	0.7930
5	Güterverkehrstochter	0.7739
6	Güter-Tochter	0.7690
7	BLS	0.7552
8	Railion	0.7550
9	Güterverkehr	0.7512
10	Gütersparte	0.7448

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Logistikunternehmen	0.8298
2	Logistikkonzern	0.8048
3	Speditions-	0.7560
4	Logistikfirma	0.7522
5	Logistik-	0.7456
6	Logistikdienstleister	0.7442
7	Logistikgruppe	0.7302
8	Logistikgeschäft	0.7198
9	Logistiksparte	0.7102
10	Panalpina	0.6681

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Frachtgeschäft	0.7955
2	Europaverkehr	0.7931
3	Cargo-Geschäft	0.7536
4	Cargo-Bereich	0.7396
5	Frachtverkehr	0.7367
6	Inlandverkehr	0.7328
7	Frachtbereich	0.7245
8	Passagierverkehr	0.7047
9	Europa-Verkehr	0.6932
10	Airline	0.6781

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Transportvolumen	0.8008
2	Frachtrate	0.7831
3	Frachtaufkommen	0.7614
4	Frachtvolumen	0.7539
5	Containergeschäft	0.7206
6	Frachttransport	0.7191
7	Passagierzahl	0.6771
8	Passagieraufkommen	0.6614
9	Luftfracht	0.6548
10	Frachtgeschäft	0.6509

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	BLS	0.7937
2	SBB_BLS	0.7901
3	Bahnunternehmen	0.7666
4	Trenitalia	0.7493
5	BLS_Lötschbergbahn	0.7355
6	RAlpin	0.7353
7	TX_Logistik	0.7312
8	SBB_Cargo	0.7297
9	Bundesbahn	0.7296
10	SBB	0.7239

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Gütermenge	0.7574
2	Verkehrsleistung	0.7401
3	Tonnenkilometer	0.7399
4	Verkehrsvolumen	0.7377
5	Transportmenge	0.7332
6	Transportleistung	0.7325
7	Tonnage	0.7279
8	Transportvolumen	0.7182
9	Containerverkehr	0.7059
10	transportiert_Menge	0.6993

TABLE B.9: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for *Güterverkehr (Freights Traffic)* in the semantic space of the word2vec model, ordered by cosine similarity

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Airline	0.8959
2	Fluggesellschaft	0.8784
3	Air_Berlin	0.8351
4	Billigflieger	0.8351
5	Lufthansa	0.8344
6	Billigfluggesellschaft	0.8283
7	Singapore_Airline	0.8175
8	Germanwings	0.8170
9	Fluglinie	0.8136
10	Billig-Airline	0.7952

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Piste_28	0.8503
2	gekröpft_Nordanflug	0.8322
3	Südanflug	0.8322
4	Ostanflug	0.8120
5	Flughafen_Zürich	0.8027
6	Piste_34	0.7958
7	Nordanflug	0.7909
8	Instrumentenlandesystem	0.7847
9	Gekröpfte	0.7832
10	Flughafen_Kloten	0.7811

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	A320	0.8257
2	Typ_Airbus	0.8254
3	A380	0.8215
4	Langstreckenflugzeug	0.8142
5	Airbus_A320	0.8137
6	Airbus	0.8093
7	Flugzeug	0.8070
8	Grossraumflugzeug	0.7941
9	Typ_A320	0.7919
10	Airbus_A380	0.7916

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Regionalpilot	0.8382
2	Kabinenpersonal	0.8292
3	Aeropers	0.8163
4	ehemalig_Crossair-Pilot	0.8014
5	Swiss_Pilots	0.7999
6	Swiss_European	0.7963
7	Bodenpersonal	0.7805
8	Ex-Crossair-Pilot	0.7644
9	Airbus-Pilot	0.7621
10	Fluggesellschaft_Swiss	0.7617

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Lufthansa	0.8613
2	British_Airways	0.8569
3	Air_Berlin	0.8533
4	Airline	0.8389
5	Fluggesellschaft	0.8293
6	Air_France	0.8291
7	KLM	0.8023
8	Iberia	0.8011
9	Air_France-KLM	0.7995
10	American_Airlines	0.7946

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Flugzeug	0.8296
2	Kleinflugzeug	0.8159
3	Passagiermaschine	0.8018
4	Typ_Cessna	0.7908
5	Boeing_737	0.7897
6	Passagierflugzeug	0.7879
7	Landeanflug	0.7876
8	Cessna	0.7860
9	Propellermaschine	0.7731
10	MD-83	0.7616

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Flughafen	0.7874
2	London_Heathrow	0.7728
3	Fluggast	0.7662
4	Airport	0.7652
5	Flughafen_Zürich	0.7482
6	Flughafen_Kloten	0.7387
7	anfliegen	0.7350
8	Zürich-Kloten	0.7277
9	Fluggesellschaft	0.7224
10	Flug	0.7178

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Flugplatz	0.8716
2	Militärflugplatz	0.8364
3	Flugplatz_Dübendorf	0.8339
4	Militärflugplatz_Dübendorf	0.7927
5	zivil_Fliegerei	0.7571
6	Dübendorfer_Militärflugplatz	0.7459
7	Dübendorfer_Flugplatz	0.7422
8	Zivilaviatik	0.7232
9	Zivilfliegerei	0.7142
10	fliegerisch_Nutzung	0.7103

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Kampfjet	0.8756
2	Kampfflugzeug	0.8413
3	Luftwaffe	0.8277
4	Flugzeug	0.7863
5	Militärflugzeug	0.7715
6	Helikopter	0.7704
7	Militärjet	0.7615
8	Transportflugzeug	0.7423
9	Lenkwaffe	0.7262
10	Drohn	0.7250

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Flugsicherung	0.8361
2	Bazl	0.8103
3	Skyguide	0.7996
4	Flugsicherung_Skyguide	0.7995
5	BAZL	0.7758
6	Flugbetrieb	0.7675
7	Zivilluftfahrt	0.7439
8	Flughafen_Zürich	0.7378
9	Flugverkehr	0.7309
10	Luftverkehr	0.7208

TABLE B.10: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for *Luftverkehr (Air Traffic)* in the semantic space of the word2vec model, ordered by cosine similarity

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Cape_Canaveral	0.8492
2	Canaveral	0.8372
3	Weltraumbahnhof_Cape	0.8298
4	US-Raumfähre_Discovery	0.8264
5	Spaceshuttle_Discovery	0.7933
6	Kourou	0.7924
7	Weltraumzentrum	0.7794
8	russisch_Sojus-Rakete	0.7782
9	Weltraumbahnhof_Baikonur	0.7768
10	Raumfähre_Atlantis	0.7761

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Raumsonde	0.8687
2	Sonde	0.8644
3	europäisch_Raumsonde	0.8294
4	rot_Planet	0.8230
5	Nasa-Sonde	0.8165
6	Hubble-Weltraumteleskop	0.8088
7	Weltraumteleskop	0.8047
8	Nasa-Raumsonde	0.8039
9	Orbiter	0.8009
10	Sonde_Rosetta	0.7983

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Raumtransporter	0.8179
2	Automated_Transfer	0.7931
3	Trägerrakete	0.7922
4	unbemannt	0.7832
5	Ariane-5-Rakete	0.7765
6	russisch_Sojus-Rakete	0.7750
7	Dragon-Kapsel	0.7661
8	Forschungssatellit	0.7648
9	Telekommunikationssatellit	0.7607
10	Raumfähre	0.7595

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Himmelskörper	0.8852
2	Sonnensystem	0.8796
3	unsre_Sonnensystem	0.8778
4	Galaxie	0.8597
5	Komet	0.8550
6	Planet	0.8432
7	Asteroid	0.8288
8	Milchstrasse	0.8238
9	unsre_Milchstrasse	0.8132
10	Astronom	0.8112

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Weltraum	0.8539
2	Raumsonde	0.8373
3	Sonde	0.8228
4	Weltall	0.8210
5	Raumschiff	0.8163
6	All	0.8112
7	Umlaufbahn	0.8058
8	Satellit	0.8058
9	Raumfahrzeug	0.8001
10	Astronaut	0.7938

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	All_befördern	0.8354
2	All_schießen	0.8240
3	Forschungssatellit	0.8098
4	All_schicken	0.8054
5	Erdumlaufbahn_bringen	0.8047
6	Mondsonde	0.7944
7	Satellit	0.7775
8	Testsatellit	0.7759
9	Raumsonde	0.7660
10	Trägerrakete	0.7653

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	international_Raumstation	0.8881
2	Raumfähre	0.8741
3	Astronaut	0.8552
4	ISS_angedockt	0.8441
5	Raumfähre_Atlantis	0.8412
6	ISS_starten	0.8374
7	Endeavour	0.8338
8	ISS_fliegen	0.8281
9	international_Weltraumstation	0.8269
10	Cape_Canaveral	0.8266

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Jupiter	0.8711
2	Mond	0.8518
3	Mars	0.8330
4	Merkur	0.8329
5	Neptun	0.8218
6	Saturn	0.8179
7	Komet	0.8165
8	sonnennah_Planet	0.7880
9	Planet	0.7869
10	Planet_Merkur	0.7786

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Langstreckenrakete	0.8283
2	ballistisch_Rakete	0.7786
3	Interkontinentalrakete	0.7644
4	ballistisch	0.7524
5	Mittelstreckenrakete	0.7517
6	Missil	0.7505
7	Marschflugkörper	0.7422
8	Nuklearsprengkopf	0.7381
9	Rakete	0.7229
10	Raketentyp	0.7196

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Nasa	0.8790
2	ESA	0.8143
3	europäisch_Raumfahrtorganisation	0.8112
4	Esa	0.8027
5	amerikanisch_Raumfahrtbehörde	0.7943
6	US-Raumfahrtbehörde_Nasa	0.7931
7	Raumsonde	0.7878
8	amerikanisch_Weltraumbehörde	0.7845
9	Jaxa	0.7842
10	europäisch_Raumfahrtagentur	0.7777

TABLE B.11: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for *Raumfahrt (Space Travel)* in the semantic space of the word2vec model, ordered by cosine similarity

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Regionalzug	0.8678
2	Personenzug	0.8317
3	Güterzug	0.8287
4	Intercity-Zug	0.8261
5	Schnellzug	0.8183
6	Eurocity-Zug	0.7613
7	Zugkomposition	0.7601
8	Intercity	0.7591
9	Regio-Express	0.7585
10	Intercityzug	0.7579

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Halbstundentakt	0.8560
2	S-Bahn	0.8503
3	Viertelstundentakt	0.8254
4	S-Bahn-Linie	0.8023
5	15-Minuten-Takt	0.8020
6	halbstündlich	0.7948
7	Buslinie	0.7894
8	Hauptverkehrszeit	0.7853
9	30-Minuten-Takt	0.7806
10	SN_7	0.7771

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Fahrgast	0.8431
2	Pendler	0.8323
3	Reisende	0.8301
4	Bahnkunde	0.7782
5	S-Bahn	0.7589
6	Bahnreisende	0.7538
7	Bahnpassagier	0.7486
8	Passagier	0.7340
9	verspätet_Zug	0.7185
10	Stosszeit	0.7079

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Bundesbahn	0.8455
2	SBB	0.8394
3	BLS	0.8330
4	Regionalverkehr	0.8092
5	Bahnunternehmen	0.8006
6	SBB_Cargo	0.7994
7	Fernverkehr	0.7829
8	Güterverkehr	0.7775
9	Personenverkehr	0.7661
10	deutsch_Bahn	0.7491

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Neubaustrecke	0.8064
2	Doppelspur	0.7752
3	Schienennetz	0.7701
4	Mattstetten-Rothrist	0.7512
5	Ost-West-Achse	0.7500
6	Bahnnetz	0.7485
7	Basistunnel	0.7445
8	S-Bahn	0.7377
9	Fernverkehr	0.7368
10	Durchmesserlinie	0.7349

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Neigezug	0.8231
2	Doppelstock	0.8148
3	Rollmaterial	0.8057
4	Fernverkehrszug	0.7992
5	Doppelstockzug	0.7985
6	S-Bahn-Zug	0.7910
7	Triebzug	0.7866
8	S-Bahn-Komposition	0.7652
9	Intercity-	0.7594
10	Regionalzug	0.7569

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Fahrzeit	0.7623
2	S-Bahn	0.7546
3	schlank_Anschluss	0.7479
4	Zug_verkehren	0.7452
5	verkehrend_Zug	0.7401
6	Nachtverbindung	0.7379
7	Fernzug	0.7330
8	Umsteigezeit	0.7282
9	Fahrplan_verkehren	0.7242
10	Fernverkehrszug	0.7211

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Haltestelle	0.8135
2	S-Bahn	0.8099
3	Tram	0.7801
4	Gleis	0.7708
5	vorzeitig_wenden	0.7677
6	Bahnhof	0.7654
7	Forchbahn	0.7563
8	Ersatzbus	0.7543
9	HB	0.7466
10	Hauptbahnhof	0.7408

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	ZVV	0.8792
2	Zürcher_Verkehrsverbund	0.8594
3	VBZ	0.8321
4	Verkehrsverbund	0.8300
5	SZU	0.7960
6	VZO	0.7914
7	Nachtnetz	0.7730
8	Verkehrsbetrieb	0.7605
9	SBB	0.7528
10	Fahrgast	0.7480

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Bahnverkehr	0.8553
2	Zugverkehr	0.8380
3	Streckenunterbruch	0.8281
4	Zugsverkehr	0.8251
5	Bahnbetrieb	0.8003
6	S-Bahn-Verkehr	0.7954
7	Bahnersatzbus	0.7773
8	Fahrleitungsstörung	0.7751
9	verkehrt_Ersatzbus	0.7722
10	Zugausfall	0.7695

TABLE B.12: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for *Schienenbverkehr/Bahn (Railway Transport/Train)* in the semantic space of the word2vec model, ordered by cosine similarity

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Schiffssteg	0.8124
2	Anlegestelle	0.8003
3	Schiffsteg	0.7726
4	Schiffsanlegestelle	0.7551
5	Ufer	0.7329
6	Landungssteg	0.7172
7	Quaimauer	0.7152
8	Steg	0.7114
9	Schiffstation	0.7106
10	Schiffsstation	0.7075

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Schifffahrt	0.8358
2	Schifffahrtsgesellschaft	0.7846
3	Schifffahrtsbetrieb	0.7032
4	Vierwaldstättersee_SGV	0.7025
5	Dampfschiff	0.6866
6	Schiffsverkehr	0.6865
7	Dampfschifffahrt	0.6723
8	Wasserweg	0.6622
9	Schifffahrtgesellschaft	0.6536
10	Rheinschifffahrt	0.6502

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Boot	0.8810
2	Segelboot	0.8576
3	Motorboot	0.8477
4	Segelschiff	0.8170
5	Jacht	0.8061
6	Ruderboot	0.7812
7	Schiff	0.7688
8	Schlauchboot	0.7607
9	Segeljacht	0.7597
10	Gummiboot	0.7564

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	kentern	0.8596
2	Seenot_geraten	0.8296
3	Fischerboot	0.8172
4	Küstenwache	0.8157
5	Seenot	0.7855
6	Küstenwacht	0.7813
7	Kentern	0.7789
8	Schlauchboot	0.7741
9	Frachter	0.7729
10	Schiff	0.7717

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Gipfelischiff	0.8560
2	Frühkurs	0.8267
3	ZSG	0.8153
4	Frühschiff	0.8147
5	Zürichsee-Schifffahrtsgesellschaft_ZSG	0.7931
6	Frühschiff_Aruf	0.7754
7	Aktion_rechtsufrig	0.7621
8	Aruf	0.7372
9	MS_Etzel	0.7257
10	Zürichsee_Schifffahrtsgesellschaft	0.7244

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Rhei	0.8230
2	MS_Panta	0.8206
3	Motorschiff_Panta	0.8072
4	Zürichsee_Schifffahrtsgesellschaft	0.8004
5	Zürichsee-_Schifffahrtsgesellschaft	0.7675
6	ZSG	0.7590
7	Öswag	0.7552
8	Panta_Rhei	0.7349
9	Zürichsee-Schifffahrtsgesellschaft_ZSG	0.7129
10	Pannenschiff	0.6429

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Zürichsee-Fähre	0.8353
2	Zürichseefähre	0.8068
3	Zürichsee-Fähre_Horgen-Meile	0.8062
4	Zürichseefähre_Horgen-Meile	0.8042
5	Fähre_Horgen-Meile	0.7627
6	Fähre	0.7011
7	ZSG-Schiff	0.6733
8	Hans_Isler	0.6446
9	Zürichsee-Schifffahrtsgesellschaft_ZSG	0.6367
10	Heinz_Blatti	0.6358

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Panta_Rhei	0.7935
2	Motorschiff	0.7602
3	Zürichsee-Schifffahrtsgesellschaft	0.7485
4	Dampfschiff	0.7464
5	Zürichsee-Schifffahrtsgesellschaft_ZSG	0.7448
6	Schiff	0.7447
7	Zürichseeschiff	0.7442
8	Dampfschiff_Stadt	0.7440
9	Zürichseeflott	0.7381
10	Raddampfer_Stadt	0.7324

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Kursschiff	0.7958
2	Fähre	0.7720
3	Schiff	0.7573
4	Motorboot	0.7414
5	Boot	0.7316
6	Anlegestelle	0.7261
7	kantonale_Seepolizei	0.7206
8	Schiffssteg	0.7089
9	Seepolizei	0.7043
10	Polizeiboot	0.7032

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Schiff	0.9003
2	Frachter	0.8756
3	Frachtschiff	0.8370
4	Tanker	0.8172
5	Kreuzfahrtschiff	0.8009
6	Fischerboot	0.7956
7	Containerschiff	0.7896
8	Passagierschiff	0.7846
9	Fischkutter	0.7734
10	Hafen	0.7673

TABLE B.13: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for *Schiffahrt (Shipping)* in the semantic space of the word2vec model, ordered by cosine similarity

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Wage	0.8489
2	Fahrzeug	0.8479
3	Auto	0.8427
4	Personenwage	0.8404
5	Lieferwagen	0.8333
6	Lastwagen	0.8170
7	Motorrad	0.7820
8	Fahrrad	0.7812
9	Sattelschlepper	0.7779
10	PW	0.7709

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Verkehrsführung	0.7938
2	Autoverkehr	0.7730
3	Verkehrsberuhigung	0.7674
4	Westtangente	0.7651
5	Veloweg	0.7633
6	Rosengartenstrass	0.7576
7	Langsamverkehr	0.7482
8	Durchgangsverkehr	0.7472
9	Umfahrung	0.7463
10	Verkehrskonzept	0.7451

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Motorradlenker	0.8945
2	Autolenkerin	0.8839
3	Autolenker	0.8751
4	Personenwagenlenker	0.8674
5	Lenkerin	0.8637
6	Rollerfahrer	0.8583
7	32-jährig_Lenker	0.8582
8	22-jährig_Lenker	0.8555
9	Lieferwagenlenker	0.8538
10	Autofahrerin	0.8537

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Kantonsstrass	0.8078
2	Autobahn	0.8021
3	zweispurig	0.7775
4	einstreifig_führen	0.7642
5	Fahrspur	0.7638
6	Fahrstreifen	0.7608
7	Gubristtunnel	0.7605
8	Fahrbahn	0.7582
9	Hauptstrasse	0.7568
10	Autobahneinfahrt	0.7542

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	öffentlich_Verkehr	0.8492
2	ÖV	0.8365
3	öffentlich_Verkehrsmittel	0.8224
4	Pendler	0.7466
5	Verkehrsmittel	0.7455
6	S-Bahn	0.7349
7	öV	0.7170
8	Bus-	0.7166
9	Tram_Bus	0.7117
10	Umsteigen	0.7047

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Fussgänger	0.8716
2	Automobilist	0.8679
3	Autofahrer	0.8649
4	Velofahrer	0.8609
5	Verkehrsteilnehmer	0.8076
6	Fahrzeuglenker	0.7946
7	Fussgängerstreife	0.7940
8	Autolenker	0.7846
9	Zebrastreifen	0.7805
10	Rotlicht	0.7559

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Radstreife	0.8476
2	Trottoir	0.8364
3	Schutzinsel	0.8168
4	Mittelinsel	0.8146
5	Lichtsignalanlage	0.8133
6	Fahrbahn	0.8110
7	Velostreife	0.8095
8	Kreisel	0.8050
9	Fussgängerübergang	0.8035
10	Radweg	0.8007

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Autoverkehr	0.8305
2	Durchgangsverkehr	0.8155
3	Mehrverkehr	0.8046
4	Verkehrsaufkommen	0.8024
5	Stau	0.8016
6	Verkehr	0.8010
7	Privatverkehr	0.7960
8	Ausweichverkehr	0.7854
9	Individualverkehr	0.7809
10	motorisiert_Verkehr	0.7773

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	S-Bahn	0.8550
2	Buslinie	0.8303
3	Haltestelle	0.7955
4	Tram	0.7703
5	Halbstundentakt	0.7653
6	Viertelstundentakt	0.7641
7	Hauptverkehrszeit	0.7610
8	15-Minuten-Takt	0.7464
9	dicht_Takt	0.7448
10	Forchbahn	0.7444

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	motorisiert_Individualverkehr	0.8441
2	Langsamverkehr	0.8033
3	Veloverkehr	0.7976
4	Individualverkehr	0.7845
5	Autoverkehr	0.7831
6	öV	0.7741
7	öffentlich_Verkehr	0.7576
8	ÖV	0.7447
9	Strassennetz	0.7440
10	Privatverkehr	0.7335

TABLE B.14: 10 most similar terms to the 10 centroids of the cluster model for the lexicon for *Strassenverkehr (Road Traffic)* in the semantic space of the word2vec model, ordered by cosine similarity

B.2 Lexical Resources for the Framing Detection Task

In this section of the Appendix, we illustrate all the lexical resources that were derived for the framing detection task. Like for the document classification resources, we choose the form of the re-embedded lexicons, represented by the centroids of the quantized lexicons, i.e., displaying the vicinities of those.

Since we have used three different embeddings (for the three languages) to derive the resources, we display them here separately.

B.2.1 German Resources

B.2.1.1 Centroids of the Lexicon for Accountability Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Verantwortung_übernehmen	0.6451
2	Konsequenz_ziehen	0.6087
3	personell_Konsequenz	0.6081
4	Verantwortung_tragen	0.5983
5	Verantwortung	0.5792
6	zurücktreten	0.5736
7	Verfehlung	0.5556
8	Fehler	0.5554
9	Hut_nehmen	0.5473
10	Fehlverhalten	0.5452

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Hut_nehmen	0.7165
2	zurücktreten	0.6793
3	Amtsantritt	0.6686
4	Nachfolger	0.6268
5	Rücktritt	0.6195
6	Abgang	0.6026
7	Weggang	0.5984
8	Handtuch_werfen	0.5964
9	Antritt	0.5877
10	schassen	0.5865

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Wahlniederlage	0.7829
2	Wahlsieg	0.7486
3	Wahlerfolg	0.6936
4	Neuwahl	0.6856
5	Wiederwahl	0.6807
6	Wahlergebnis	0.6748
7	Kandidatur	0.6675
8	Wahl	0.6627
9	Parlamentswahl	0.6578
10	Wahlschlappe	0.6547

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Kündigung	0.7744
2	Entlassung	0.6889
3	Arbeitsvertrag	0.6673
4	Arbeitsverhältnis	0.6139
5	fristlos_Kündigung	0.6123
6	Aufhebungsvertrag	0.5985
7	unbefristet_beschäftigt	0.5984
8	Sozialplan	0.5850
9	entlassen	0.5837
10	Abfindung	0.5786

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Strafe	0.7512
2	Buße	0.6930
3	Sanktion	0.6615
4	sanktionieren	0.6593
5	Verstoß	0.6588
6	Strafgeld	0.6567
7	Bußgeld	0.6506
8	Strafzahlung	0.6486
9	Geldbuße	0.6463
10	ahnden	0.6424

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	zurücktreten	0.7572
2	Amt_niederlegen	0.6825
3	gesundheitlich_Grund	0.6509
4	Rücktritt	0.6493
5	entlassen	0.6339
6	niederlegen	0.6330
7	Nachfolger	0.6312
8	Amt	0.6308
9	Hut_nehmen	0.6214
10	Amt_zurücktreten	0.6172

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Rücktritt	0.8441
2	zurücktreten	0.7478
3	Demission	0.6993
4	Absetzung	0.6905
5	Abgang	0.6864
6	Abwahl	0.6684
7	Ernennung	0.6521
8	Ausscheide	0.6479
9	Rauswurf	0.6432
10	Nachfolger	0.6394

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Kandidatur	0.7897
2	Amtsinhaber	0.7814
3	kandidieren	0.7810
4	Wiederwahl	0.7763
5	Kandidat	0.7584
6	Gegenkandidaten	0.7563
7	erst_Wahlgang	0.7504
8	Präsidentenamt	0.7480
9	wiederwählen	0.7438
10	dritt_Amtszeit	0.7038

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Nachfolger	0.7821
2	Nachfolge	0.7783
3	Stellvertreter	0.7473
4	zurücktreten	0.7403
5	Amt	0.7349
6	zurückgetreten	0.7323
7	beerben	0.7187
8	amtierend	0.7161
9	Vize	0.7096
10	ernennen	0.6965

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Kandidatur	0.6777
2	Bisherige	0.6656
3	Ersatzwahl	0.6596
4	kandidieren	0.6544
5	Stadtpräsidium	0.6503
6	SVP	0.6488
7	Kandidierende	0.6467
8	Nationalratswahl	0.6368
9	Kandidatin	0.6320
10	Nomination	0.6293

TABLE B.15: 10 most similar terms to the 10 centroids of the cluster model for the German lexicon for Accountability Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.1.2 Centroids of the Lexicon for Deliberation Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	diskutieren	0.7648
2	reden	0.6960
3	erörtern	0.6892
4	auseinandersetzen	0.6849
5	ansprechen	0.6807
6	befassen	0.6756
7	thematisieren	0.6671
8	debattieren	0.6552
9	Thema	0.6434
10	besprechen	0.6417

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	abstimmen	0.7419
2	beschlossen	0.7227
3	beschließen	0.7110
4	zustimmen	0.7107
5	darüber_abstimmen	0.6905
6	debattieren	0.6850
7	diskutieren	0.6836
8	absegnen	0.6741
9	beraten	0.6719
10	darüber_beraten	0.6623

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Debatte	0.7859
2	Diskussion	0.7850
3	Kontroverse	0.7462
4	hitzig_Debatte	0.6744
5	heftig_Debatte	0.6714
6	kontrovers	0.6689
7	Auseinandersetzung	0.6385
8	geführt_Debatte	0.6365
9	hitzig	0.6324
10	debattieren	0.6231

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Gespräch	0.7885
2	Verhandlung	0.7480
3	Treffen	0.7315
4	Gesprächsrund	0.7298
5	Verhandlungsrund	0.6820
6	Vorgespräch	0.6751
7	Sondierungsgespräch	0.6621
8	Unterredung	0.6587
9	verhandeln	0.6433
10	Einigung	0.6408

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Abkomme	0.8061
2	Vereinbarung	0.7902
3	Übereinkunft	0.7304
4	Abkommen	0.6958
5	aushandeln	0.6878
6	unterzeichnen	0.6847
7	Vertrag	0.6818
8	Abmachung	0.6714
9	vereinbaren	0.6698
10	Kooperation	0.6660

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Stellung_nehmen	0.7041
2	ausarbeiten	0.6496
3	präzisieren	0.6465
4	vorlegen	0.6390
5	konkretisieren	0.6273
6	erörtern	0.6229
7	abklären	0.6214
8	vorliegen	0.6146
9	unterbreiten	0.6086
10	weit_Vorgehen	0.6085

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Kontakt	0.7165
2	zusammenarbeiten	0.7068
3	eng_Kontakt	0.6416
4	eng_Zusammenarbeit	0.6388
5	Dialog	0.6229
6	Zusammenarbeit	0.6207
7	Kooperation	0.6205
8	direkt_Kontakt	0.6130
9	persönlich_Kontakt	0.6099
10	Einvernehmen	0.6060

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	überprüfen	0.7657
2	prüfen	0.7628
3	klären	0.7524
4	untersuchen	0.7383
5	abklären	0.7211
6	analysieren	0.6404
7	beurteilen	0.6361
8	prüfen_ob	0.6267
9	durchleuchten	0.6221
10	erörtern	0.6219

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	informieren	0.7270
2	kontaktieren	0.6981
3	anrufen	0.6965
4	telefonisch	0.6614
5	darüber_informieren	0.6398
6	übermittelt	0.6187
7	weiterleiten	0.6126
8	anfordern	0.6096
9	bitten	0.6082
10	erfahren	0.6078

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Streit	0.8183
2	streiten	0.7667
3	heftig_Streit	0.6983
4	Auseinandersetzung	0.6765
5	ringen	0.6752
6	Disput	0.6745
7	Streit_um	0.6733
8	Zwist	0.6724
9	Konflikt	0.6666
10	Debatte	0.6637

TABLE B.16: 10 most similar terms to the 10 centroids of the cluster model for the German lexicon for Deliberation Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.1.3 Centroids of the Lexicon for Efficacy Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	ungenügend	0.7666
2	mangelhaft	0.7592
3	unzureichend	0.7355
4	unbefriedigend	0.6815
5	miserabel	0.6776
6	schlecht	0.6417
7	unzulänglich	0.6337
8	dürftig	0.6271
9	zufriedenstellend	0.6188
10	lückenhaft	0.6047

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	erfüllen	0.7409
2	umsetzen	0.7104
3	einhalten	0.6999
4	Umsetzung	0.6849
5	Erfüllung	0.6532
6	nachkommen	0.6530
7	Vorgabe	0.6521
8	verbindlich	0.6139
9	Einhaltung	0.5908
10	festlegen	0.5888

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	gescheitert	0.7447
2	scheitern	0.6969
3	missglückt	0.6737
4	misslingen	0.6674
5	Versuch	0.6499
6	Fehlschlag	0.6394
7	Scheit	0.6209
8	glücken	0.6134
9	geglückt	0.6064
10	misslingt	0.6062

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Sieg	0.7728
2	Triumph	0.7092
3	Niederlage	0.7062
4	siegen	0.6936
5	antreten	0.6858
6	Sieger	0.6601
7	besiegen	0.6510
8	triumphieren	0.6494
9	gewinnen	0.6492
10	erringen	0.6413

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	unterzeichnen	0.7968
2	ratifizieren	0.7910
3	Abkomme	0.7883
4	Ratifizierung	0.7858
5	Unterzeichnung	0.7726
6	Vereinbarung	0.7298
7	Ratifikation	0.7071
8	Vertragswerk	0.7044
9	unterzeichnet	0.6930
10	unterschreiben	0.6924

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Desaster	0.8244
2	Katastrophe	0.7254
3	Debakel	0.7202
4	Versagen	0.7132
5	verheerend	0.6689
6	Fiasko	0.6630
7	katastrophal	0.6623
8	Fehler	0.6496
9	Fehlentscheidung	0.6458
10	Misere	0.6322

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	ausreichend	0.8015
2	notwendig	0.7769
3	nötig	0.7652
4	genügend	0.7442
5	vorhanden	0.7435
6	erforderlich	0.7348
7	geeignet	0.7070
8	benötigen	0.7017
9	ausreichen	0.6628
10	eignen	0.6581

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	steigern	0.7547
2	erzielen	0.6594
3	verbessern	0.6541
4	Steigerung	0.6424
5	dank	0.6144
6	übertreffen	0.5983
7	zulegen	0.5979
8	erhöhen	0.5955
9	verbessert	0.5951
10	erreichen	0.5941

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Niederlage	0.7659
2	Rückschlag	0.7352
3	Schlappe	0.7094
4	Scheit	0.6470
5	Enttäuschung	0.6431
6	Debakel	0.6425
7	herb_Niederlage	0.6383
8	schwer_Niederlage	0.6248
9	herb_Rückschlag	0.6230
10	Desaster	0.6117

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	schaffen	0.7964
2	gelingen	0.7764
3	brauchen	0.7586
4	helfen	0.7413
5	doch	0.7380
6	erreichen	0.7272
7	bringen	0.7267
8	fehlen	0.7212
9	gewinnen	0.7210
10	eben	0.7188

TABLE B.17: 10 most similar terms to the 10 centroids of the cluster model for the German lexicon for Efficacy Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.1.4 Centroids of the Lexicon for Efficiency Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Kosten_sparen	0.6983
2	Einsparung	0.6916
3	Effizienzgewinn	0.6821
4	Synergie_nutzen	0.6439
5	sparen	0.6324
6	Kosteneinsparung	0.6226
7	Kostensenkung	0.6210
8	Kosten	0.6187
9	Effizienzsteigerung	0.6184
10	einsparen	0.6110

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	wirkungsvoll	0.7723
2	effektiv	0.7631
3	wirksam	0.7609
4	zielgerichtet	0.6381
5	konsequent	0.6222
6	rigoros	0.6214
7	Instrument	0.6168
8	präventiv	0.6063
9	effizient	0.6036
10	Maßnahme	0.5935

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	weiterentwickeln	0.7168
2	Weiterentwicklung	0.7154
3	leistungsfähig	0.7097
4	neuartig	0.6710
5	innovativ	0.6200
6	entwickeln	0.5944
7	effizient	0.5681
8	Technologie	0.5674
9	Steuerungstechnologie	0.5641
10	Membrantechnologie	0.5597

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Wirtschaftlichkeit	0.7054
2	Wirksamkeit	0.6950
3	Qualität	0.6821
4	Effizienz	0.6787
5	Effektivität	0.6488
6	Nutzen	0.6232
7	Leistungsfähigkeit	0.6164
8	patientenrelevant	0.6006
9	Qualitätssicherung	0.5990
10	Verbesserung	0.5893

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	geschickt	0.7513
2	intelligent	0.7261
3	clever	0.7213
4	raffiniert	0.7173
5	klug	0.7046
6	brillant	0.6949
7	schlau	0.6940
8	genial	0.6834
9	simpel	0.6455
10	witzig	0.6434

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	betrieblich	0.6175
2	Leistungserstellung	0.5791
3	Leistungsbereitstellung	0.5643
4	Gewinnzuschlag	0.5590
5	Doppik	0.5570
6	Ausführungswahrscheinlichkeit	0.5554
7	Mitteleinsatz	0.5539
8	Leistungsrechnung	0.5534
9	betriebswirtschaftlich	0.5422
10	Leistungserbringung	0.5414

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	effizient	0.7571
2	kostengünstig	0.7524
3	optimal	0.6890
4	optimieren	0.6668
5	flexibel	0.6271
6	bedarfsgerecht	0.6260
7	möglichst_effizient	0.6182
8	effektiv	0.6166
9	bestmöglich	0.6136
10	umweltverträglich	0.5995

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	umweltfreundlich	0.8549
2	umweltschonend	0.8090
3	effizient	0.7806
4	klimafreundlich	0.7800
5	umweltverträglich	0.7725
6	energieeffizient	0.7410
7	ressourcenschonend	0.7373
8	nachhaltig	0.7129
9	klimaschonend	0.6916
10	kostengünstig	0.6858

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	entschlossen	0.7088
2	energisch	0.6955
3	kompromisslos	0.6745
4	konsequent	0.6711
5	hart	0.6604
6	entschieden	0.6549
7	zielstrebig	0.6394
8	unerbittlich	0.6360
9	mutig	0.6219
10	beharrlich	0.6175

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	kompetent	0.7297
2	verantwortungsvoll	0.7209
3	verantwortungsbewusst	0.6905
4	umsichtig	0.6889
5	vernünftig	0.6886
6	klug	0.6882
7	seriös	0.6733
8	pragmatisch	0.6521
9	glaubwürdig	0.6512
10	professionell	0.6508

TABLE B.18: 10 most similar terms to the 10 centroids of the cluster model for the German lexicon for Efficiency Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.1.5 Centroids of the Lexicon for Epistemic Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Forscher	0.8790
2	Wissenschaftler	0.8420
3	Biologe	0.7841
4	Physiker	0.7441
5	Forscherteam	0.7184
6	Forschergruppe	0.6723
7	Forscherin	0.6485
8	Wissenschafter	0.6473
9	Paläontologe	0.6272
10	Cox-Foster	0.6181

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Studie	0.8338
2	Untersuchung	0.7162
3	Analyse	0.6929
4	Langzeitstudie	0.6688
5	Auswertung	0.6603
6	veröffentlicht_Studie	0.6578
7	aktuell_Studie	0.6546
8	Forschungsergebnis	0.6372
9	Übersichtsartikel	0.6358
10	Befund	0.6338

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Journal_of	0.7746
2	Annals_of	0.7590
3	Fachmagazin_Journal	0.7523
4	Internal_Medicine	0.7388
5	Fachjournal	0.7325
6	Psychiatry	0.7294
7	Medicine_online	0.7115
8	Plos	0.7108
9	Neurology	0.7100
10	Fachblatt	0.7089

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	ETH	0.7021
2	ETH_Zürich	0.6936
3	Forschungsprojekt	0.6713
4	wissenschaftlich	0.6708
5	Universität_Bern	0.6473
6	Neuroinformatik	0.6409
7	interdisziplinär	0.6313
8	Universität_Zürich	0.6298
9	Wissenschaft	0.6289
10	Forschungsarbeit	0.6286

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Professor	0.8580
2	Universität_Zürich	0.8175
3	Lehrstuhl	0.8115
4	Politikwissenschaft	0.7849
5	Dozent	0.7758
6	Universität_Bern	0.7721
7	Universität	0.7614
8	Volkswirtschaftslehre	0.7551
9	Professorin	0.7517
10	Universität_Freiburg	0.7400

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Forscher	0.7352
2	Fachblatt_Science	0.7087
3	Fachmagazin_PNAS	0.6882
4	Fachjournal_Nature	0.6870
5	Fachblatt_Nature	0.6852
6	Medicine_online	0.6826
7	Fachzeitschrift_Nature	0.6783
8	Wissenschaftsmagazin_Nature	0.6752
9	Fachmagazin_Nature	0.6705
10	Geoscience	0.6694

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Studie	0.7330
2	veröffentlicht_Studie	0.7210
3	aktuell_Studie	0.6951
4	repräsentativ_Umfrage	0.6821
5	Auftrag_gegeben	0.6730
6	erstellt_Studie	0.6600
7	repräsentativ_Befragung	0.6499
8	durchgeführt_Studie	0.6467
9	Marktforschungsinstitut	0.6435
10	Psephos	0.6361

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	ETH_Zürich	0.7846
2	Umweltphysik	0.7610
3	Universität_Bern	0.7455
4	Universität_Zürich	0.7263
5	Meteorologie_präsidiert	0.7248
6	Klimaphysik	0.7185
7	geographisch_Institut	0.7130
8	Atmosphären	0.6991
9	Polymerforschung	0.6955
10	Nutztierwissenschaft	0.6936

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Universität_Zürich	0.8432
2	Universität_Basel	0.7893
3	Universität_Freiburg	0.7826
4	Professor	0.7656
5	Wirtschaftsrecht	0.7621
6	Verfassungsvergleichung	0.7589
7	vergleichend_Verfassungsrecht	0.7496
8	Lehrstuhl	0.7471
9	emeritiert_Professor	0.7470
10	Titularprofessor	0.7440

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Weltklimarat	0.7911
2	IPCC	0.7892
3	IPCC-Bericht	0.7383
4	Klimaforscher	0.7255
5	Weltklimarat_IPCC	0.7188
6	Wissenschaftler	0.7176
7	Wissenschafter	0.7070
8	Klimarat	0.6891
9	Klimaforschung	0.6802
10	Fachleute	0.6647

TABLE B.19: 10 most similar terms to the 10 centroids of the cluster model for the German lexicon for Epistemic Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.1.6 Centroids of the Lexicon for Legality Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	EU-Recht	0.6613
2	rechtlich	0.6598
3	Regelung	0.6532
4	Verbot	0.6329
5	Verfahren	0.6255
6	Vorschrift	0.6226
7	Vorratsdaten-Richtlinie	0.6208
8	Rechtsprechung	0.6198
9	Gericht	0.6183
10	Recht	0.6158

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Korruptionsskandal	0.7905
2	Korruptionsaffäre	0.7545
3	Korruptionsvorwurf	0.7442
4	Korruptionsfall	0.6909
5	Affäre	0.6906
6	Korruption	0.6636
7	Bestechung	0.6462
8	Schmiergeldzahlung	0.6337
9	angeblich_Luxusreis	0.6181
10	Titelmissbrauch	0.6139

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	anklagen	0.8410
2	wegen_Betrug	0.7558
3	wegen_Steuerhinterziehung	0.7500
4	Anklage	0.7465
5	Angeklagte	0.7355
6	verurteilen	0.7259
7	Jahr_Haft	0.7214
8	freisprechen	0.7174
9	angeklagt	0.7169
10	Anklage_gegen	0.7132

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Mafia	0.8031
2	'_Ndrangheta	0.7421
3	Camorra	0.7345
4	'Ndrangheta	0.7287
5	Sacra_Corona	0.7160
6	Ndrangheta	0.7050
7	Cosa_Nostra	0.6983
8	Mafiosi	0.6889
9	Kalabrien	0.6856
10	Clan	0.6679

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Untreue	0.7887
2	Bestechung	0.7758
3	Betrug	0.7739
4	Urkundenfälschung	0.7623
5	Bestechlichkeit	0.7597
6	Veruntreuung	0.7288
7	Betrug_Urkundenfälschung	0.7246
8	anklagen	0.7195
9	Amtsmissbrauch	0.7149
10	Steuerhinterziehung	0.7117

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Drogenhandel	0.8630
2	organisiert_Kriminalität	0.8403
3	organisiert_Verbrechen	0.7741
4	Kriminalität	0.7697
5	Menschenhandel	0.7680
6	Geldwäsche	0.7577
7	kriminell	0.7412
8	Drogenschmuggel	0.7398
9	Waffenhandel	0.7274
10	Schmuggel	0.7191

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Schmiergeld	0.7800
2	Schmiergeldzahlung	0.7217
3	schwarz_Kasse	0.6899
4	Bestechung	0.6848
5	Bestechungsgeld	0.6807
6	Machenschaft	0.6361
7	dubios	0.6330
8	kriminell_Machenschaft	0.6262
9	Betrug	0.6255
10	kriminell	0.6236

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Korruption	0.8179
2	korrupt	0.7922
3	Vetternwirtschaft	0.7547
4	Misswirtschaft	0.6844
5	Staatsapparat	0.6753
6	autoritär	0.6417
7	Nepotismus	0.6382
8	korrupt_Politiker	0.6321
9	Klientelismus	0.6248
10	politisch_Klasse	0.6217

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Terrorist	0.8120
2	Terror	0.7982
3	terroristisch	0.7937
4	Terrorismus	0.7887
5	Terrorgruppe	0.7860
6	al-Qaida	0.7602
7	Extremist	0.7564
8	islamistisch_Terrorist	0.7400
9	Terrororganisation	0.7398
10	islamistisch_Terror	0.7357

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Drogenkartell	0.8994
2	organisiert_Verbrechen	0.8328
3	Drogenmafia	0.8243
4	Zeta	0.8213
5	Drogenband	0.8173
6	Mafia	0.8058
7	Drogenhandel	0.7983
8	Drogenhändler	0.7966
9	Drogenboss	0.7803
10	Drogenbaron	0.7634

TABLE B.20: 10 most similar terms to the 10 centroids of the cluster model for the German lexicon for Legality Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.1.7 Centroids of the Lexicon for Participation Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Stimmberechtigte	0.7872
2	Gemeindeversammlung	0.7858
3	Stimmvolk	0.7593
4	Urnenabstimmung	0.7409
5	befinden_og	0.7385
6	Stimmbürger	0.7231
7	Projektierungskredit	0.7156
8	Entwicklungskonzept_Zeughausareal	0.6887
9	Gemeindeparlament	0.6733
10	allfällig_Einspracheverhandlung	0.6711

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Gegenvorschlag	0.8233
2	Volksinitiative	0.7971
3	indirekt_Gegenvorschlag	0.7535
4	Initianten	0.7354
5	Volksbegehren	0.7067
6	Volkssouveränität_statt	0.6830
7	direkt_Gegenvorschlag	0.6772
8	Landschaftsinitiative	0.6680
9	parlamentarisch_Initiative	0.6640
10	Verfassungsartikel	0.6632

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Nationalratswahl	0.7999
2	Regierungsratswahl	0.7201
3	Gemeinderatswahl	0.7174
4	Gemeindewahl	0.7166
5	eidgenössisch_Wahl	0.7145
6	Grossratswahl	0.6974
7	Gesamterneuerungswahl	0.6931
8	Parlamentswahl	0.6723
9	Kommunalwahl	0.6642
10	Kantonsratswahl	0.6636

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Parlamentswahl	0.8869
2	Präsidentenwahl	0.8363
3	Präsidentschaftswahl	0.8244
4	Wahl	0.8156
5	Urnengang	0.8054
6	Kommunalwahl	0.7854
7	Wahlergebnis	0.7716
8	Neuwahl	0.7543
9	Stichwahl	0.7372
10	vorgezogen_Wahl	0.7355

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Volksabstimmung	0.8783
2	Referendum	0.8676
3	Nein	0.7783
4	EU-Verfassung	0.7727
5	Abstimmung	0.7591
6	Volksbefragung	0.7491
7	Volksentscheid	0.7166
8	Ja	0.7026
9	Plebiszit	0.6973
10	Votum	0.6911

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	darüber_abstimmen	0.6825
2	abstimmen	0.6815
3	Parlament	0.6750
4	Abstimmung	0.6717
5	Volksvertreter	0.6710
6	zustimmen	0.6672
7	Parlamentarier	0.6601
8	Bürger	0.6585
9	Abgeordnete	0.6473
10	Volk	0.6397

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Stimmberechtigte	0.8502
2	Stimmbürger	0.8094
3	Stimmende	0.8068
4	Stimmvolk	0.8063
5	Ja-Stimme	0.7732
6	Urne	0.7360
7	Stimmbeteiligung	0.7260
8	Ja-Stimmen-Anteil	0.6927
9	Dreiviertelmehr	0.6900
10	Ja-Anteil	0.6870

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Ja-Parole	0.7001
2	Stimmvolk	0.6983
3	Nein-Parole	0.6961
4	Stimmberechtigte	0.6842
5	Ja-Parole_beschlossen	0.6695
6	SVP	0.6693
7	Referendum_ergreifen	0.6679
8	Ja-Parole_fassen	0.6591
9	Abstimmungsvorlage	0.6575
10	Behördenreferendum	0.6567

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	SVP-Initiative	0.7555
2	Masseneinwanderungsinitiative	0.7354
3	Volksinitiative	0.7051
4	Stimmvolk	0.7042
5	Ausschaffungsinitiative	0.7037
6	gegen_Masseneinwanderung	0.6866
7	Volksabstimmung	0.6835
8	Ecopop-Initiative	0.6800
9	Gegenvorschlag	0.6777
10	Personenfreizügigkeit	0.6640

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Volksbegehren	0.7749
2	Volksinitiative	0.7552
3	Begehren	0.6804
4	Unterschriftensammlung	0.6771
5	Gegenvorschlag	0.6633
6	Volksentscheid	0.6512
7	Umweltverband_linken	0.6462
8	Initianten	0.6448
9	Pistenmoratorium	0.6221
10	Behördeninitiative	0.6210

TABLE B.21: 10 most similar terms to the 10 centroids of the cluster model for the German lexicon for Participation Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.1.8 Centroids of the Lexicon for Representation Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Repräsentantenhaus	0.8173
2	Republikaner	0.8109
3	Demokrat	0.7973
4	Senat	0.7881
5	Senator	0.7727
6	Harry_Reid	0.7277
7	Kongress	0.7188
8	republikanisch	0.7104
9	weiß_Haus	0.7039
10	US-Senat	0.6944

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	SVP	0.8443
2	CVP	0.8412
3	Grünliberale	0.8342
4	GLP	0.8320
5	SP	0.8319
6	BDP	0.8093
7	Freisinnige	0.7983
8	EVP	0.7746
9	Bürgerliche	0.7717
10	EDU	0.7672

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Minister	0.8560
2	Wirtschaftsminister	0.7697
3	Innenminister	0.7511
4	Umweltminister	0.7310
5	Staatssekretär	0.7262
6	Finanzminister	0.7172
7	Arbeitsminister	0.7041
8	Ressortchef	0.7022
9	Ministerpräsident	0.7019
10	Außenminister	0.7006

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Partei	0.8445
2	Sozialdemokrat	0.8428
3	Linke	0.8128
4	Liberale	0.7991
5	Regierungspartei	0.7977
6	Koalitionspartner	0.7977
7	Linkspartei	0.7811
8	Parteichef	0.7769
9	Koalition	0.7724
10	Volkspartei	0.7723

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Premier	0.8648
2	Premierminister	0.8314
3	Staatspräsident	0.8031
4	Ministerpräsident	0.7971
5	Oppositionsführer	0.7681
6	Staatschef	0.7583
7	Parlamentspräsident	0.7354
8	Parteichef	0.7330
9	Regierungschef	0.7226
10	Außenminister	0.7189

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	SPD	0.8914
2	FDP	0.8665
3	groß_Koalition	0.8367
4	CDU	0.8148
5	Grüne	0.8115
6	CDU_CSU	0.7994
7	Union	0.7832
8	Rot-Grün	0.7822
9	schwarz-gelb_Koalition	0.7759
10	Schwarz-Gelb	0.7728

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Ständerat	0.8951
2	Nationalrat	0.8601
3	klein_Kammer	0.8117
4	Herbstsession	0.7561
5	vorberatend_Kommission	0.7453
6	Sommersession	0.7399
7	Rechtskommission	0.7388
8	eidgenössisch_Rat	0.7357
9	als_Erstrat	0.7211
10	Wintersession	0.7131

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Parlament	0.8693
2	Abgeordnete	0.8289
3	Parlamentarier	0.7614
4	Volksvertretung	0.7140
5	Volksvertreter	0.7136
6	Nationalversammlung	0.6985
7	beide_Kammer	0.6889
8	Abgeordnetenhaus	0.6875
9	Unterhaus	0.6784
10	Fraktion	0.6689

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Kantonsrat	0.8327
2	Regierungsrat	0.7936
3	Kantonsparlament	0.7743
4	bürgerlich_Mehrheit	0.7245
5	SVP-Fraktion	0.7036
6	SP	0.6999
7	Stadtrat	0.6933
8	Gros_Rat	0.6860
9	Zürcher_Kantonsrat	0.6827
10	Stadtparlament	0.6819

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Vizepremier	0.7767
2	Premier	0.7269
3	Premierminister	0.7227
4	stellvertretend_Ministerpräsident	0.7215
5	Vizeregierungschef	0.6915
6	Minister	0.6767
7	Ministerpräsident	0.6617
8	Verteidigungsminister	0.6536
9	Parlamentspräsident	0.6462
10	Staatspräsident	0.6451

TABLE B.22: 10 most similar terms to the 10 centroids of the cluster model for the German lexicon for Representation Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.1.9 Centroids of the Lexicon for Stakeholder Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	Hilfsorganisation	0.7924
2	UNHCR	0.7196
3	humanitär_Hilfe	0.7162
4	IKRK	0.6788
5	international_Hilfsorganisation	0.6725
6	WFP	0.6652
7	Welternährungsprogramm	0.6643
8	rot_Kreuz	0.6613
9	vereint_Nation	0.6557
10	UN	0.6546

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	Pro_Natura	0.7942
2	Vogelschutz_WWF	0.7046
3	Stiftung_Landschaftsschutz	0.6739
4	pro_Natura	0.6621
5	WWF	0.6569
6	VCS	0.6330
7	Schweizer_Vogelschutz	0.6290
8	Umweltorganisation	0.6276
9	Umweltverband	0.6183
10	WWF_VCS	0.6176

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	Lobbyist	0.6990
2	Wirtschaftsverband	0.6894
3	Interessengruppe	0.6812
4	Interessenvertreter	0.6672
5	Nichtregierungsorganisation	0.6617
6	Lobbygruppe	0.6521
7	NGO	0.6293
8	Vertreter	0.6227
9	Lobby	0.6155
10	Gewerkschaft	0.6095

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	Nichtregierungsorganisation	0.7227
2	Hilfswerke	0.7084
3	Entwicklungsorganisation	0.6970
4	Hilfsorganisation	0.6885
5	NGO	0.6781
6	Heks	0.6640
7	Hilfswerk	0.6562
8	Alliance_Sud	0.6352
9	Misereor	0.6248
10	Amnesty_International	0.6231

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Uno	0.8085
2	vereint_Nation	0.7966
3	UNO	0.7889
4	international_Organisation	0.7773
5	UN	0.7452
6	UN-Organisation	0.6671
7	Weltbank	0.6568
8	Sonderorganisation	0.6473
9	Uno-Organisation	0.6275
10	Staatengemeinschaft	0.6250

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Landschaftsschutz	0.7141
2	Landschaftsschützer	0.6682
3	Naturschützer	0.6618
4	Umweltverband	0.6521
5	Pro_Natura	0.6282
6	Naturschutz	0.6280
7	Rodewald	0.5975
8	Vogelschützer	0.5942
9	Vogelschutz	0.5928
10	Naturschutzverband	0.5900

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Umweltorganisation	0.8606
2	Umweltverband	0.8446
3	WWF	0.8239
4	Umweltschutzorganisation	0.8239
5	Greenpeace	0.8229
6	Umweltschützer	0.8200
7	Naturschützer	0.7360
8	Umweltgruppe	0.7226
9	Umweltorganisation_Greenpeace	0.7143
10	Naturschutzorganisation	0.7107

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	Amnesty_International	0.8603
2	Menschenrechtsorganisation	0.8594
3	Menschenrechtler	0.7993
4	Menschenrechtsgruppe	0.7895
5	Human_Rights	0.7888
6	Amnesty	0.7698
7	Watch	0.7542
8	Menschenrechtsorganisation_Human	0.7417
9	Rights_Watch	0.7169
10	HRW	0.7138

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	Dachverband	0.7577
2	Branchenverband	0.7498
3	Verband	0.7373
4	Indiesuisse	0.7020
5	komplementärmedizinisch_Heilmittel	0.6859
6	schweizerisch_alpwirtschaftlich	0.6800
7	Fachverband	0.6495
8	Vereinigung	0.6438
9	Interessenvertretung	0.6393
10	Spielwarenindustrie_DVSI	0.6323

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	Nichtregierungsorganisation	0.8192
2	NGO	0.7888
3	Menschenrechtsorganisation	0.7622
4	Organisation	0.7118
5	Menschenrechtsgruppe	0.6873
6	Zivilgesellschaft	0.6810
7	nichtstaatlich_Organisation	0.6804
8	Amnesty_International	0.6663
9	Menschenrechtler	0.6660
10	Aktivist	0.6650

TABLE B.23: 10 most similar terms to the 10 centroids of the cluster model for the German lexicon for Stakeholder Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.1.10 Centroids of the Lexicon for Transparency Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	geheim	0.7114
2	vertraulich	0.7112
3	Indiskretion	0.6511
4	Dokument	0.6499
5	publik	0.6448
6	zuspielen	0.6421
7	enthüllen	0.6378
8	publik_machen	0.6369
9	Enthüllung	0.6239
10	brisant	0.6217

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	dokumentieren	0.7733
2	untersuchen	0.6969
3	analysieren	0.6896
4	aufklären	0.6574
5	aufarbeiten	0.6510
6	auswerten	0.6469
7	detailliert	0.6322
8	zusammentragen	0.6315
9	anhand	0.6276
10	recherchieren	0.6245

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	erklären	0.8100
2	betonen	0.7811
3	bestätigen	0.7794
4	behaupten	0.7448
5	berichten	0.7366
6	versichern	0.7308
7	erfahren	0.7205
8	wissen	0.7110
9	meinen	0.7084
10	sagen	0.7042

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	publizieren	0.7620
2	veröffentlicht	0.7566
3	publiziert	0.7408
4	verfassen	0.7088
5	Publikation	0.6951
6	veröffentlichen	0.6913
7	ausführlich	0.6796
8	herausgeben	0.6566
9	Dokument	0.6529
10	vorliegend	0.6475

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Daten	0.7924
2	persönlich_Daten	0.7126
3	Information	0.7036
4	sensibel_Daten	0.6946
5	Datensatz	0.6822
6	personenbezogen_Daten	0.6669
7	Kundendaten	0.6651
8	Zahlungsinformation	0.6607
9	PNR-Zentralstelle	0.6450
10	gespeichert_Daten	0.6425

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	Detail	0.7215
2	vorlegen	0.7167
3	konkret	0.7094
4	detailliert	0.6943
5	prüfen	0.6866
6	diskutieren	0.6850
7	klären	0.6727
8	umfassend	0.6457
9	informieren	0.6452
10	Einzelheit	0.6352

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	verschleiern	0.7321
2	verschweigen	0.7221
3	vertuschen	0.7036
4	verheimlichen	0.6400
5	Verschleierung	0.6116
6	suggerieren	0.6034
7	kaschieren	0.5967
8	unterschlagen	0.5967
9	entlarven	0.5961
10	vorwerfen	0.5927

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	informieren	0.7243
2	Auskunft	0.6943
3	kontaktieren	0.6557
4	Unterlage	0.6283
5	anfordern	0.6271
6	darüber_informieren	0.6255
7	Stellung_nehmen	0.6111
8	weiterleiten	0.5974
9	abklären	0.5891
10	erkundigen	0.5867

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	transparent	0.7702
2	Transparenz	0.7146
3	offenlegen	0.6708
4	intransparent	0.6554
5	mehr_Transparenz	0.6250
6	nachvollziehbar	0.6237
7	Intransparenz	0.6045
8	Offenlegung	0.5977
9	undurchsichtig	0.5889
10	objektiv	0.5829

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	darauf_hinweisen	0.7410
2	einsehen	0.6940
3	eingestehen	0.6824
4	klarmachen	0.6780
5	daran_erinnern	0.6484
6	Eindruck_erwecken	0.6484
7	darüber_informieren	0.6468
8	klarstellen	0.6419
9	davon_ausgehen	0.6214
10	zugeben	0.6046

TABLE B.24: 10 most similar terms to the 10 centroids of the cluster model for the German lexicon for Transparency Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.2 English Resources

B.2.2.1 Centroids of the Lexicon for Accountability Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	resign	0.8036
2	resignation	0.7954
3	sack	0.7642
4	sacking	0.7633
5	quit	0.7517
6	dismissal	0.7254
7	departure	0.6776
8	appoint	0.6389
9	appointment	0.6358
10	oust	0.6312

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	oversight	0.7351
2	supervision	0.7066
3	monitoring	0.7028
4	supervise	0.6931
5	inspection	0.6634
6	examination	0.6576
7	auditing	0.6469
8	supervisory	0.6352
9	internal_audit	0.6213
10	monitor	0.6208

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	honesty	0.8344
2	objectivity	0.7908
3	impartiality	0.7568
4	professionalism	0.7530
5	probity	0.7099
6	sincerity	0.7048
7	decency	0.6847
8	fairness	0.6810
9	trustworthiness	0.6790
10	integrity	0.6655

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	decision-making	0.8161
2	decision-make	0.7569
3	decision-making_process	0.7485
4	decisionmaking	0.7232
5	decision_making	0.7201
6	policymaking	0.7107
7	policy-making	0.6943
8	policymake	0.6380
9	decisionmake	0.6264
10	accountability	0.6132

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	stricter	0.7823
2	regulation	0.7575
3	strict	0.7391
4	stringent	0.7190
5	enforce	0.6965
6	tougher	0.6612
7	rule	0.6611
8	toughen	0.6449
9	flout	0.6237
10	draconian	0.6208

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	oust	0.8753
2	topple	0.8621
3	overthrow	0.8551
4	ouster	0.8246
5	ousting	0.7935
6	depose	0.7659
7	toppling	0.7595
8	unseat	0.7519
9	unseated	0.6802
10	overthrown	0.6786

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	abide	0.7589
2	adhere	0.7394
3	adherence	0.7224
4	comply	0.6992
5	strict_compliance	0.6968
6	conformity	0.6941
7	strictly_adhere	0.6661
8	basic_principle	0.6600
9	enshrine	0.6531
10	strictly_observe	0.6512

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	reappoint	0.8415
2	re-elect	0.8028
3	appoint	0.7638
4	nominate	0.7428
5	re-appoint	0.7357
6	elect	0.6980
7	re-election	0.6961
8	seek_re-election	0.6922
9	reelect	0.6843
10	re-appointed	0.6832

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	transparency	0.7824
2	transparency_accountability	0.7759
3	accountability	0.7573
4	accountability_transparency	0.7241
5	consistency	0.7093
6	predictability	0.6899
7	openness	0.6884
8	governance	0.6713
9	openness_transparency	0.6697
10	fairness	0.6393

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	negligence	0.7340
2	wrongdoing	0.7261
3	prosecute	0.7227
4	hold_accountable	0.6695
5	prosecution	0.6668
6	misconduct	0.6617
7	dishonesty	0.6578
8	cover-up	0.6557
9	negligent	0.6525
10	wrongdo	0.6491

TABLE B.25: 10 most similar terms to the 10 centroids of the cluster model for the English lexicon for Accountability Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.2.2 Centroids of the Lexicon for Deliberation Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	insist	0.8324
2	admit	0.8276
3	acknowledge	0.8049
4	argue	0.7762
5	suggest	0.7667
6	confirm	0.7398
7	reveal	0.7397
8	warn	0.7220
9	explain	0.7160
10	believe	0.7138

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	closed-door_session	0.7366
2	closed_session	0.7363
3	closed-door	0.7247
4	closed-door_meeting	0.6923
5	meeting	0.6829
6	deliberation	0.6749
7	preparatory_meeting	0.6726
8	closed-door_talk	0.6609
9	closed_door	0.6559
10	closed-door_discussion	0.6116

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	mutual_understanding	0.7807
2	pragmatic_cooperation	0.7781
3	cooperation	0.7520
4	mutually_beneficial	0.7499
5	multifaceted_cooperation	0.7402
6	bilateral_relation	0.7256
7	mutually-beneficial_cooperation	0.7201
8	people-to-people_exchange	0.7189
9	traditional_friendship	0.7186
10	deepen_pragmatic	0.7108

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	seminar	0.8095
2	round-table_discussion	0.7942
3	roundtable_discussion	0.7772
4	round_table	0.7267
5	roundtable	0.6780
6	round-table_debate	0.6473
7	symposium	0.6445
8	workshop	0.6372
9	round-table	0.5956
10	forum	0.5521

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	debate	0.7987
2	heated_debate	0.6985
3	lively_debate	0.6352
4	discussion	0.6182
5	polemic	0.5981
6	heated_discussion	0.5840
7	discourse	0.5599
8	conversation	0.5563
9	argument	0.5529
10	heated	0.5466

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	ask	0.8214
2	speak	0.7584
3	tell	0.7526
4	explain	0.7040
5	talk	0.7002
6	question	0.6976
7	comment	0.6761
8	describe	0.6759
9	discuss	0.6715
10	respond	0.6682

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	dialogue	0.7789
2	constructive_dialogue	0.7778
3	constructive	0.6868
4	constructive_engagement	0.6705
5	cooperation	0.6320
6	mutually_beneficial	0.6224
7	constructive_interaction	0.6130
8	sincere	0.5850
9	pragmatic	0.5793
10	structured_dialogue	0.5784

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	spat	0.7760
2	disagreement	0.7532
3	dispute	0.7474
4	bitter_dispute	0.7391
5	squabble	0.7371
6	tussle	0.7349
7	wrangle	0.7256
8	quarrel	0.7225
9	stand-off	0.7203
10	feud	0.7197

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	controversial	0.6478
2	debatable	0.6267
3	contentious	0.6084
4	hotly_debate	0.6071
5	hotly_debated	0.6057
6	disputable	0.6004
7	divisive	0.5403
8	emotive	0.5230
9	contentious_issue	0.5181
10	much_debated	0.5125

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	squabble	0.8437
2	wrangle	0.8277
3	bicker	0.8228
4	feud	0.7904
5	wrangling	0.7683
6	bickering	0.7679
7	infighting	0.6903
8	recrimination	0.6759
9	infight	0.6714
10	stalemate	0.6706

TABLE B.26: 10 most similar terms to the 10 centroids of the cluster model for the English lexicon for Deliberation Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.2.3 Centroids of the Lexicon for Efficacy Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	emission	0.8201
2	greenhouse_gas	0.7934
3	carbon_emission	0.7565
4	pollution	0.7299
5	carbon_dioxide	0.7249
6	co2_emission	0.7084
7	greenhouse-gas_emission	0.6912
8	emitter	0.6830
9	carbon_pollution	0.6780
10	greenhouse	0.6718

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	growth	0.7004
2	economy	0.6986
3	recovery	0.6910
4	sluggish	0.6530
5	job_creation	0.6512
6	economic	0.6377
7	slowdown	0.6281
8	weak	0.6254
9	consumer_spending	0.6188
10	confidence	0.6103

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	success	0.7758
2	achievement	0.7305
3	successful	0.6950
4	progress	0.6522
5	achieve	0.6341
6	accomplishment	0.6286
7	impressive	0.6194
8	experience	0.6079
9	effort	0.6009
10	remarkable	0.5938

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	solve	0.7639
2	problem	0.7222
3	tackle	0.7175
4	address	0.7042
5	resolve	0.6828
6	answer	0.6435
7	serious	0.6401
8	challenge	0.6127
9	solution	0.5936
10	silver_bullet	0.5744

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	ineffective	0.8129
2	incompetent	0.7251
3	inept	0.7053
4	useless	0.6982
5	ineffectual	0.6980
6	inadequate	0.6840
7	impractical	0.6584
8	flawed	0.6507
9	irresponsible	0.6496
10	inefficient	0.6382

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	good	0.7486
2	really	0.7293
3	indeed	0.7229
4	though	0.7162
5	actually	0.7104
6	just	0.7073
7	great	0.7038
8	so	0.7006
9	then	0.6958
10	again	0.6946

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	debacle	0.7249
2	calamity	0.7156
3	disastrous	0.7069
4	catastrophic	0.7051
5	disaster	0.6874
6	fiasco	0.6844
7	collapse	0.6835
8	meltdown	0.6782
9	crisis	0.6487
10	chaos	0.6466

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	both	0.6407
2	and	0.6405
3	however	0.6338
4	also	0.6321
5	the	0.6203
6	plan	0.6178
7	development	0.6169
8	that	0.6132
9	support	0.6121
10	these	0.6079

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	easy	0.6983
2	impossible	0.6975
3	difficult	0.6578
4	vital	0.6525
5	unlikely	0.6428
6	essential	0.6423
7	important	0.6344
8	able	0.6258
9	crucial	0.6093
10	obvious	0.6054

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	disappointing	0.6878
2	strong	0.6840
3	weak	0.6733
4	drop	0.6680
5	decline	0.6648
6	result	0.6464
7	solid	0.6454
8	lackluster	0.6387
9	gain	0.6356
10	rise	0.6307

TABLE B.27: 10 most similar terms to the 10 centroids of the cluster model for the English lexicon for Efficacy Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.2.4 Centroids of the Lexicon for Efficiency Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	efficient	0.8416
2	environmentally_friendly	0.8084
3	energy-efficient	0.7636
4	environmentally-friendly	0.7604
5	cost-effective	0.7597
6	cost-efficient	0.7484
7	eco-friendly	0.7077
8	highly_efficient	0.7055
9	cleaner	0.6932
10	greener	0.6831

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	development	0.6735
2	business	0.6576
3	support	0.6329
4	economic	0.6223
5	global	0.6178
6	project	0.6153
7	government	0.5955
8	and	0.5942
9	country	0.5928
10	the	0.5927

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	budget_deficit	0.8144
2	deficit	0.7900
3	budget	0.7851
4	spending	0.7462
5	spending_cut	0.7147
6	expenditure	0.6840
7	deficit_reduction	0.6839
8	overspend	0.6761
9	budgetary	0.6644
10	public_finances	0.6619

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	borrow	0.6931
2	borrowing	0.6877
3	funding	0.6849
4	cash	0.6718
5	short-term	0.6632
6	debt	0.6451
7	money	0.6302
8	savings	0.6177
9	fund	0.6174
10	loan	0.6099

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	mismanagement	0.6737
2	inefficiency	0.6616
3	problem	0.6577
4	chaos	0.6292
5	failure	0.6172
6	incompetence	0.6161
7	corruption	0.5881
8	delay	0.5876
9	rampant_corruption	0.5871
10	paralysis	0.5845

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	quick	0.6838
2	good	0.6702
3	again	0.6418
4	though	0.6399
5	little	0.6325
6	indeed	0.6312
7	pace	0.6312
8	then	0.6281
9	great	0.6274
10	decent	0.6268

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	increase	0.7177
2	cost	0.6905
3	reduce	0.6517
4	benefit	0.6445
5	income	0.6366
6	overall	0.6236
7	revenue	0.6096
8	pay	0.6055
9	improve	0.6021
10	boost	0.5971

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	inefficient	0.7848
2	wasteful	0.7741
3	ineffective	0.7245
4	costly	0.7099
5	bureaucratic	0.6850
6	unnecessary	0.6640
7	burdensome	0.6622
8	bureaucracy	0.6560
9	unproductive	0.6495
10	inept	0.6371

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	approach	0.6457
2	however	0.6275
3	plan	0.6087
4	must	0.6044
5	yet	0.6022
6	way	0.5980
7	course	0.5967
8	programme	0.5925
9	process	0.5924
10	effort	0.5911

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	efficiency	0.8021
2	operational_efficiency	0.7452
3	optimize	0.7266
4	optimization	0.7222
5	optimise	0.7007
6	profitability	0.6639
7	productivity	0.6266
8	optimisation	0.6262
9	rationalization	0.6238
10	incremental	0.6135

TABLE B.28: 10 most similar terms to the 10 centroids of the cluster model for the English lexicon for Efficiency Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.2.5 Centroids of the Lexicon for Epistemic Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	scholar	0.7993
2	thinker	0.7914
3	historian	0.7853
4	philosopher	0.7693
5	theorist	0.7446
6	sociologist	0.7138
7	author	0.6948
8	university_professor	0.6785
9	academic	0.6634
10	eminent	0.6604

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	biology	0.8083
2	genetics	0.7622
3	neuroscience	0.7616
4	physiology	0.7441
5	molecular_biology	0.7429
6	microbiology	0.7304
7	cell_biology	0.7303
8	cognitive_neuroscience	0.7103
9	molecular	0.6997
10	developmental_biology	0.6959

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	sociology	0.8044
2	phd	0.7995
3	faculty	0.7957
4	physics	0.7747
5	mathematics	0.7735
6	university	0.7664
7	doctorate	0.7658
8	zoology	0.7645
9	anthropology	0.7494
10	lecturer	0.7478

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	study	0.7272
2	research	0.6948
3	researcher	0.6905
4	science	0.6714
5	expert	0.6615
6	research_institute	0.6515
7	institute	0.6255
8	academic	0.6173
9	applied	0.6149
10	analysis	0.5988

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	biologist	0.7543
2	scientist	0.7494
3	physicist	0.7134
4	anthropologist	0.7023
5	researcher	0.6913
6	geologist	0.6829
7	psychologist	0.6797
8	marine_biologist	0.6624
9	neuroscientist	0.6606
10	geneticist	0.6564

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	graduate	0.8146
2	undergraduate	0.8133
3	postgraduate	0.7846
4	university	0.7841
5	student	0.7612
6	college	0.7498
7	mathematics	0.7277
8	phd	0.7246
9	scholarship	0.7244
10	diploma	0.7182

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	professor	0.8792
2	oxford_university	0.8284
3	cambridge_university	0.8216
4	harvard	0.8020
5	university	0.7935
6	columbium_university	0.7927
7	harvard_university	0.7911
8	lecturer	0.7873
9	stanford_university	0.7793
10	senior_lecturer	0.7736

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	ipcc	0.8319
2	intergovernmental_panel	0.8239
3	pachaurus	0.7664
4	rajendra_pachaurus	0.7442
5	intergovernment_panel	0.7202
6	ipcc_rajendra	0.6988
7	ipcc_intergovernmental	0.6790
8	nobel-winning_intergovernmental	0.6754
9	climate_change	0.6730
10	global_warming	0.6574

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	report	0.6248
2	community	0.6236
3	global	0.5991
4	climate	0.5929
5	environmental	0.5861
6	also	0.5832
7	initiative	0.5756
8	environment	0.5749
9	economic	0.5721
10	that	0.5715

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	londondr	0.7492
2	liverpoolprofessor	0.7442
3	hepatology	0.7376
4	reproductive_biology	0.7342
5	epidemiology	0.7231
6	baylor_college	0.7151
7	professor	0.7149
8	psychiatry	0.7064
9	john_hopkin	0.7045
10	m.d_m.p.h	0.7030

TABLE B.29: 10 most similar terms to the 10 centroids of the cluster model for the English lexicon for Epistemic Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.2.6 Centroids of the Lexicon for Legality Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	wrongdoing	0.7870
2	wrongdo	0.7454
3	misconduct	0.7433
4	bribery	0.7357
5	allegation	0.7273
6	fraud	0.7219
7	malpractice	0.7210
8	alleged	0.7086
9	cover-up	0.7053
10	negligence	0.6899

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	torture	0.7916
2	atrocity	0.7844
3	rape	0.7569
4	summary_execution	0.7547
5	extrajudicial_killing	0.7498
6	enforce_disappearance	0.7429
7	rape_torture	0.7428
8	mass_murder	0.7357
9	sexual_violence	0.7357
10	arbitrary_arrest	0.7349

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	corruption	0.8595
2	rampant_corruption	0.7608
3	nepotism	0.7581
4	cronyism	0.7538
5	graft	0.7179
6	mismanagement	0.7047
7	corruption_nepotism	0.7041
8	incompetence	0.7030
9	endemic_corruption	0.6891
10	corruption_mismanagement	0.6511

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	imprisonment	0.8463
2	jail	0.8425
3	sentence	0.8045
4	prison_sentence	0.7987
5	imprison	0.7761
6	convict	0.7740
7	behind_bar	0.7726
8	prison	0.7684
9	jail_sentence	0.7634
10	sentencing	0.7397

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	supreme_court	0.6900
2	constitutional_court	0.6589
3	constitutionality	0.6484
4	court	0.6404
5	judicial	0.6394
6	ruling	0.6244
7	constitutional	0.6210
8	tribunal	0.6114
9	appellate_court	0.6109
10	legality	0.6090

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	plaintiff	0.8066
2	lawsuit	0.7921
3	defendant	0.7593
4	attorney	0.7487
5	lawyer	0.7382
6	lawsuit_file	0.7333
7	court	0.7102
8	juror	0.6939
9	class-action_lawsuit	0.6918
10	judge	0.6885

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	brutality	0.7848
2	despicable	0.7426
3	shameful	0.7338
4	barbaric	0.7296
5	cruelty	0.7292
6	savagery	0.7284
7	heinous	0.7161
8	vile	0.7081
9	shameless	0.7068
10	barbarism	0.7027

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	murder	0.8080
2	killing	0.7991
3	kidnap	0.7566
4	kidnapping	0.7522
5	slaying	0.7314
6	murderer	0.7210
7	abduction	0.7183
8	abduct	0.7077
9	rape	0.7046
10	behead	0.6961

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	convict	0.8389
2	plead_guilty	0.7608
3	felony	0.7559
4	find_guilty	0.7451
5	conspiracy	0.7322
6	acquit	0.7266
7	offence	0.7254
8	criminal	0.7228
9	insider_trading	0.7221
10	perjury	0.7170

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	drug_trafficking	0.8518
2	human_trafficking	0.8434
3	trafficking	0.8088
4	organized_crime	0.7908
5	smuggling	0.7866
6	drug_smuggling	0.7829
7	organize_crime	0.7788
8	illegal_migration	0.7663
9	organised_crime	0.7631
10	narcotic	0.7586

TABLE B.30: 10 most similar terms to the 10 centroids of the cluster model for the English lexicon for Legality Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.2.7 Centroids of the Lexicon for Participation Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	initiative	0.6660
2	commitment	0.6637
3	support	0.6154
4	process	0.5950
5	demand	0.5868
6	strategy	0.5766
7	and	0.5587
8	effort	0.5512
9	system	0.5488
10	employ_533,000	0.5450

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	government	0.7404
2	public	0.6803
3	national	0.6750
4	power	0.6146
5	president	0.6138
6	state	0.5746
7	support	0.5388
8	that	0.5357
9	say	0.5345
10	leader	0.5339

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	civil_society	0.8150
2	non-governmental_organization	0.7608
3	nongovernmental_organization	0.7231
4	ngo	0.7171
5	non_governmental	0.6824
6	non-governmental_organisation	0.6782
7	non-government_organization	0.6766
8	inter-governmental_non-governmental	0.6645
9	civic_society	0.6569
10	non-governmental	0.6277

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	election	0.7384
2	vote	0.7138
3	voter	0.6974
4	campaign	0.6845
5	mayoral	0.6374
6	candidate	0.6336
7	ballot	0.6317
8	re-election	0.6058
9	referendum	0.6056
10	electoral	0.5809

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	advocacy	0.7707
2	advocacy_group	0.7669
3	nonprofit	0.7171
4	nonpartisan_nonprofit	0.7026
5	nonpartisan	0.6884
6	ngo	0.6113
7	non-government_organisation	0.6085
8	washington-based	0.6040
9	nonprofit_organization	0.5923
10	non-governmental_organization	0.5872

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	engage	0.7606
2	participate	0.7067
3	actively_participate	0.6965
4	actively_engage	0.6961
5	active	0.6500
6	organize	0.6303
7	actively	0.6023
8	active_participation	0.5825
9	initiate	0.5609
10	partake	0.5578

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	democracy	0.7614
2	democratic	0.6864
3	constitution	0.6632
4	constitutional	0.6244
5	participatory_democracy	0.6087
6	legitimacy	0.5985
7	pluralism	0.5895
8	political	0.5755
9	self-government	0.5656
10	democratization	0.5642

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	chance	0.7421
2	right	0.7172
3	course	0.7103
4	matter	0.6536
5	opportunity	0.6407
6	yet	0.6388
7	though	0.6369
8	really	0.6356
9	indeed	0.6334
10	always	0.6330

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	engagement	0.7155
2	stakeholder	0.6929
3	active_participation	0.6778
4	participation	0.6408
5	interaction	0.6299
6	dialogue	0.6258
7	cooperation	0.6099
8	consultation	0.5853
9	involvement	0.5791
10	collaboration	0.5625

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	community	0.7656
2	civic	0.6613
3	local	0.6543
4	citizen	0.6086
5	resident	0.6068
6	citizenry	0.5921
7	grassroot	0.5805
8	administer_community-level	0.5751
9	council	0.5748
10	grass-roots	0.5656

TABLE B.31: 10 most similar terms to the 10 centroids of the cluster model for the English lexicon for Participation Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.2.8 Centroids of the Lexicon for Representation Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	state	0.6765
2	authority	0.6655
3	government	0.6586
4	region	0.6234
5	local	0.6234
6	support	0.6184
7	community	0.6134
8	national	0.6084
9	country	0.6027
10	council	0.6006

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	swearing-in_ceremony	0.8107
2	swear	0.8024
3	sworn-in	0.7745
4	swearing-in	0.7256
5	cabinet_reshuffle	0.6851
6	swearing	0.6114
7	oath	0.5934
8	reshuffle	0.5482
9	president-elect	0.5329
10	inauguration	0.5324

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	conservative	0.8191
2	mp	0.7478
3	tory	0.7400
4	party	0.7197
5	labour	0.7145
6	coalition	0.6936
7	opposition	0.6852
8	backbencher	0.6784
9	liberal	0.6640
10	politician	0.6628

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	reappoint	0.7944
2	nominate	0.7674
3	appoint	0.7638
4	re-elect	0.7629
5	elect	0.7192
6	nomination	0.7103
7	re-appoint	0.6850
8	re-election	0.6752
9	appointment	0.6719
10	resign	0.6719

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	decision	0.6376
2	leadership	0.5906
3	support	0.5871
4	meeting	0.5821
5	proposal	0.5762
6	member	0.5738
7	initiative	0.5596
8	strategy	0.5468
9	statement	0.5428
10	executive	0.5428

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	deputy_speaker	0.7389
2	parliament	0.7261
3	upper_house	0.7218
4	parliamentarian	0.6935
5	speaker	0.6904
6	parliamentary	0.6794
7	upper_chamber	0.6738
8	newly_elect	0.6612
9	newly-elected	0.6599
10	parliament_speaker	0.6389

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	lawmaker	0.7879
2	senator	0.7574
3	legislator	0.7423
4	legislature	0.7381
5	democrat	0.7338
6	congress	0.7204
7	senate	0.7134
8	republican	0.6970
9	congressman	0.6825
10	congressional	0.6728

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	democratically_elect	0.8439
2	elected	0.8052
3	depose	0.7541
4	overthrow	0.7418
5	topple	0.7028
6	democratically-elected	0.6995
7	oust	0.6832
8	ouster	0.6474
9	unseat	0.6449
10	popularly_elect	0.6437

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	presidential_candidate	0.8380
2	presidential_race	0.8132
3	presidential_hopeful	0.7578
4	presidential_election	0.7491
5	presidential_nominee	0.7399
6	presidential_nomination	0.7330
7	presidential_contender	0.7182
8	barack_obama	0.7039
9	republican	0.6970
10	candidate	0.6919

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	civil_society	0.7222
2	ngo	0.7028
3	non-governmental_organization	0.6887
4	businesspeople	0.6648
5	non-governmental_organisation	0.6345
6	non-government_organization	0.6259
7	organization	0.6190
8	nongovernmental_organization	0.6182
9	civil-society	0.6041
10	organisation	0.5912

TABLE B.32: 10 most similar terms to the 10 centroids of the cluster model for the English lexicon for Representation Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.2.9 Centroids of the Lexicon for Stakeholder Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	tony_bosworth	0.7446
2	tony_juniper	0.7095
3	roger_higman	0.7053
4	nick_rau	0.7010
5	earth	0.6937
6	julian_kirby	0.6882
7	campaigner	0.6855
8	simon_bullock	0.6848
9	andy_atkin	0.6833
10	greenpeace	0.6822

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	seminar	0.7906
2	roundtable	0.7624
3	round_table	0.7595
4	forum	0.7452
5	symposium	0.7159
6	roundtable_discussion	0.6870
7	workshop	0.6786
8	round-table_discussion	0.6708
9	round-table	0.6293
10	conference	0.6122

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	community	0.6363
2	support	0.6328
3	company	0.6275
4	business	0.6241
5	group	0.6239
6	also	0.6183
7	and	0.6148
8	development	0.6013
9	both	0.6004
10	industry	0.5974

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	partnership	0.7515
2	collaboration	0.7451
3	engagement	0.7069
4	partner	0.6491
5	work_collaboratively	0.6082
6	collaborate	0.6014
7	expertise	0.5888
8	initiative	0.5718
9	collaborative	0.5700
10	relationship	0.5657

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	chairman	0.7185
2	director	0.6939
3	executive	0.6808
4	chief	0.6753
5	deputy	0.6615
6	spokesman	0.6261
7	adviser	0.6247
8	head	0.6212
9	president	0.6200
10	chair	0.6121

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	wildlife	0.7871
2	rspb	0.7853
3	wwf	0.7795
4	conservationist	0.7103
5	species	0.6964
6	birdlife	0.6955
7	iucn	0.6936
8	wildlife_trust	0.6923
9	greenpeace	0.6563
10	conservation	0.6355

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	greenpeace	0.7755
2	wwf	0.7408
3	oxfam	0.7067
4	gerd_leipold	0.6904
5	campaigner	0.6386
6	ian_leggett	0.6326
7	oisin_coghlan	0.6302
8	walhus_friend	0.6210
9	tove_ryding	0.6201
10	kathrin_gutmann	0.6165

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	environmentalist	0.7846
2	activist	0.7350
3	campaigner	0.7309
4	environmental_campaigner	0.7181
5	lobby	0.6905
6	green_campaigner	0.6730
7	ngo	0.6171
8	conservationist	0.6114
9	ecologist	0.5996
10	politician	0.5905

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	nonprofit	0.7955
2	non-governmental_organization	0.7765
3	nonprofit_organization	0.7605
4	ngo	0.7550
5	non-governmental_organisation	0.7370
6	non-profit	0.7257
7	non-profit_organization	0.7143
8	charity	0.7009
9	organization	0.6975
10	nongovernmental_organization	0.6945

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	civil_society	0.8193
2	non-governmental_organization	0.7346
3	ngo	0.7150
4	non-government_organization	0.6860
5	non-governmental_organisation	0.6725
6	nongovernmental_organization	0.6545
7	inter-governmental_non-governmental	0.6411
8	active_participation	0.6296
9	nongovernmental	0.6268
10	non-government	0.6263

TABLE B.33: 10 most similar terms to the 10 centroids of the cluster model for the English lexicon for Stakeholder Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.2.10 Centroids of the Lexicon for Transparency Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	provide	0.7422
2	accessible	0.7337
3	available	0.7028
4	easily_accessible	0.6968
5	reliable	0.6674
6	access	0.6236
7	information	0.6196
8	comprehensive	0.5973
9	up-to-date	0.5835
10	convenient	0.5713

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	secretive	0.7180
2	secret	0.7002
3	mysterious	0.6892
4	shadowy	0.6855
5	shady	0.6717
6	obscure	0.6653
7	conceal	0.6420
8	hide	0.6092
9	secrecy	0.5875
10	secrecy_surround	0.5726

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	transparency	0.8260
2	transparency_accountability	0.7736
3	accountability_transparency	0.7350
4	openness_transparency	0.7305
5	openness	0.7145
6	fairness	0.6944
7	honesty	0.6904
8	accountability	0.6836
9	greater_transparency	0.6811
10	predictability	0.6679

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	transparent	0.8202
2	straightforward	0.6702
3	predictable	0.6579
4	comprehensible	0.6485
5	responsive	0.6415
6	simpler	0.6407
7	coherent	0.6356
8	streamlined	0.6319
9	fairer	0.6252
10	proactive	0.6233

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	opaque	0.7260
2	undemocratic	0.7214
3	non-transparent	0.7009
4	dubious	0.6771
5	dishonest	0.6625
6	cynical	0.6441
7	questionable	0.6429
8	irresponsible	0.6351
9	anti-democratic	0.6341
10	hypocritical	0.6328

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	unbiased	0.7252
2	thoughtful	0.6991
3	impartial	0.6923
4	honest	0.6922
5	thorough	0.6874
6	scrupulous	0.6843
7	truthful	0.6664
8	pragmatic	0.6574
9	candid	0.6504
10	informed	0.6421

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	auditing	0.7474
2	audit	0.7261
3	reporting	0.6743
4	monitoring	0.6651
5	oversight	0.6613
6	scrutiny	0.6490
7	internal_audit	0.5905
8	disclosure	0.5869
9	transparency	0.5783
10	accountability	0.5724

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	deception	0.8087
2	dishonest	0.7695
3	deceit	0.7489
4	fraudulent	0.7023
5	deceitful	0.6806
6	blatant	0.6676
7	dubious	0.6630
8	manipulative	0.6583
9	shameless	0.6535
10	improper	0.6520

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	decision-making	0.8622
2	decision-make	0.8169
3	decision-making_process	0.8110
4	decisionmaking	0.7631
5	decision_making	0.7155
6	governance	0.6497
7	democratic_legitimacy	0.6492
8	policymaking	0.6357
9	accountability	0.6276
10	policy-making	0.6271

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	supervise	0.7804
2	oversight	0.7717
3	supervision	0.7646
4	supervisory	0.7128
5	supervisor	0.6842
6	supervisory_authority	0.6379
7	oversee	0.6172
8	overseer	0.5860
9	systemic_risk	0.5840
10	regulatory_framework	0.5697

TABLE B.34: 10 most similar terms to the 10 centroids of the cluster model for the English lexicon for Transparency Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.3 French Resources

B.2.3.1 Centroids of the Lexicon for Accountability Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	responsabilité	0.7401
2	coresponsabilité	0.5880
3	irresponsabilité	0.5792
4	position	0.5569
5	conséquence	0.5542
6	assumer	0.5293
7	responsabiliser	0.5086
8	dominant	0.5052
9	vis-à-vis_de	0.4890
10	clairement	0.4807

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	sanctionner	0.7643
2	condamnation	0.7584
3	infliger	0.7512
4	condamner	0.7199
5	amende	0.7179
6	sanction	0.6752
7	écoper	0.6298
8	442,5_million	0.6171
9	épingler	0.6118
10	amende_record	0.5964

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	porter_plainte	0.8394
2	plainte	0.8392
3	déposer_plainte	0.8239
4	déposer	0.7772
5	plainte_déposer	0.7404
6	contre_X	0.6893
7	recours	0.6742
8	intenter	0.5850
9	débouter	0.5639
10	saisir	0.5597

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	régulation	0.7722
2	supervision	0.7710
3	surveillance	0.7126
4	réglementation	0.7029
5	contrôle	0.6823
6	régulateur	0.6077
7	grandement_tributaire	0.5520
8	ARR	0.5425
9	mutuelle_Acam	0.5284
10	Arcep_ex-ART	0.5277

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	démissionner	0.7999
2	démission	0.7611
3	limoger	0.7595
4	démettre	0.7173
5	limogeage	0.6726
6	destituer	0.6584
7	démissionnaire	0.6424
8	éviction	0.6133
9	congédier	0.6086
10	destitution	0.5792

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	dénoncer	0.7474
2	affirmer	0.7339
3	déclarer	0.7180
4	expliquer	0.7177
5	souligner	0.6951
6	interroger	0.6937
7	rappeler	0.6867
8	ajouter	0.6864
9	estimer	0.6843
10	assurer	0.6686

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	enquête	0.8051
2	investigation	0.7997
3	enquêteur	0.7756
4	enquêter	0.7526
5	commission_rogatoire	0.6802
6	instruction	0.6135
7	enquête_préliminaire	0.6112
8	cosaisi	0.5930
9	l'enquête	0.5927
10	Jean-Louis_Périè	0.5810

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	tribunal	0.7798
2	procédure	0.7623
3	audition	0.6764
4	juge	0.6728
5	Tribunal	0.6693
6	procès	0.6655
7	instruction	0.6539
8	justice	0.6398
9	audience	0.5977
10	tribunal_correctionnel	0.5862

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	adopter	0.7195
2	ratifier	0.6804
3	soumettre	0.6364
4	voter	0.6136
5	approuver	0.6128
6	vote	0.5895
7	adoption	0.5774
8	ratification	0.5528
9	amender	0.5258
10	Waxman-Markey	0.5184

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	soupçonner	0.8067
2	accuser	0.7750
3	mettre_en_cause	0.7553
4	accusation	0.7311
5	suspecter	0.6693
6	corruption	0.6587
7	présumer	0.6422
8	abus	0.6265
9	complicité	0.6059
10	reprocher	0.5996

TABLE B.35: 10 most similar terms to the 10 centroids of the cluster model for the French lexicon for Accountability Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.3.2 Centroids of the Lexicon for Deliberation Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	intransigeance	0.7768
2	inflexible	0.7394
3	extrême	0.6746
4	fermeté	0.5781
5	intransigeant	0.5544
6	attitude	0.4800
7	détermination	0.4625
8	brutalité	0.4581
9	inflexibilité	0.4536
10	habileté	0.4464

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	accord	0.7546
2	négociation	0.7432
3	compromis	0.7203
4	discussion	0.6800
5	dialogue	0.6788
6	négocier	0.6646
7	pourparler	0.6495
8	arrangement	0.5992
9	concertation	0.5940
10	accord-cadre	0.5603

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	discuter	0.7713
2	débattre	0.6690
3	discussion	0.6492
4	dialoguer	0.6476
5	négocier	0.5897
6	d'accord	0.5852
7	aborder	0.5828
8	contact	0.5692
9	dialogue	0.5632
10	négociation	0.5463

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	querelle	0.7484
2	divergence	0.7325
3	dispute	0.7263
4	différend	0.7153
5	désaccord	0.6948
6	dissension	0.6846
7	controverse	0.6803
8	contentieux	0.6781
9	rivalité	0.6720
10	conflit	0.6644

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	dire	0.8042
2	voir	0.7544
3	vouloir	0.7503
4	savoir	0.7465
5	parler	0.7442
6	toujours	0.7365
7	penser	0.7324
8	entendre	0.7311
9	poursuivre	0.7238
10	alors	0.7238

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	idée	0.6803
2	argument	0.6542
3	discours	0.6421
4	argumentation	0.6370
5	réponse	0.6143
6	raisonnement	0.6123
7	argumentaire	0.6103
8	propos	0.5948
9	position	0.5628
10	méthode	0.5528

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	tractation	0.7888
2	âpre_négociation	0.7235
3	discussion	0.6961
4	négociation	0.6934
5	pourparler	0.6896
6	intense_tractation	0.6404
7	âpre_discussion	0.6334
8	laborieux_négociation	0.6095
9	laborieux_tractation	0.6072
10	d'intenses_tractation	0.6044

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	débat	0.6480
2	politique	0.6469
3	enfin	0.5912
4	enquête	0.5837
5	démocratie	0.5766
6	justice	0.5639
7	déclaration	0.5621
8	donc	0.5595
9	citoyen	0.5520
10	car	0.5473

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	accepter	0.6703
2	rejeter	0.6700
3	décision	0.6596
4	proposition	0.6509
5	décider	0.6144
6	conseil	0.6065
7	délibération	0.5952
8	demande	0.5801
9	refuser	0.5791
10	approuver	0.5765

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	réunion	0.8338
2	rencontre	0.8266
3	conférence	0.7167
4	sommet	0.7128
5	entrevue	0.6894
6	rendez-vous	0.6831
7	rencontre_informel	0.6384
8	déjeuner	0.6279
9	table_rond	0.6166
10	dîner	0.6116

TABLE B.36: 10 most similar terms to the 10 centroids of the cluster model for the French lexicon for Deliberation Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.3.3 Centroids of the Lexicon for Efficacy Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	problème	0.6913
2	difficulté	0.6911
3	insuffisance	0.6517
4	manque	0.6104
5	faiblesse	0.5949
6	dysfonctionnement	0.5852
7	crise	0.5829
8	défaillance	0.5552
9	absence	0.5547
10	incapacité	0.5537

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	baisse	0.7348
2	diminuer	0.7288
3	augmenter	0.7280
4	diminution	0.7273
5	réduire	0.7247
6	réduction	0.7142
7	hausse	0.6864
8	augmentation	0.6796
9	croissance	0.6480
10	baisser	0.6320

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	conséquence	0.7503
2	impact	0.7184
3	répercussion	0.6816
4	engendrer	0.6459
5	affecter	0.6349
6	impact_négatif	0.6347
7	entraîner	0.6300
8	incidence	0.6206
9	catastrophique	0.6031
10	aggraver	0.6008

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	débâcle	0.7698
2	déroute	0.7482
3	désastre	0.7216
4	effondrement	0.6778
5	fiasco	0.6669
6	déconfiture	0.6614
7	faillite	0.6576
8	naufrage	0.6326
9	catastrophe	0.6325
10	échec	0.6113

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	satisfaisant	0.7222
2	efficace	0.6864
3	convaincant	0.6792
4	fiable	0.6685
5	réaliste	0.6266
6	performant	0.6219
7	pertinent	0.6189
8	crédible	0.6183
9	encourageant	0.6126
10	solide	0.6125

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	efficacité	0.6940
2	crédibilité	0.6128
3	réussite	0.5940
4	progrès	0.5883
5	qualité	0.5760
6	succès	0.5591
7	amélioration	0.5533
8	cohérence	0.5512
9	fiabilité	0.5512
10	capacité	0.5458

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	réussir	0.7050
2	capable	0.6996
3	essayer	0.6235
4	besoin	0.6188
5	suffire	0.6178
6	parvenir	0.6177
7	incapable	0.5953
8	capacité	0.5854
9	volonté	0.5847
10	vraiment	0.5839

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	attendre	0.6846
2	donc	0.6795
3	car	0.6726
4	enfin	0.6445
5	demander	0.6368
6	prévoir	0.6351
7	imposer	0.6327
8	pourtant	0.6326
9	poursuivre	0.6295
10	alors_que	0.6286

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	accord	0.8412
2	protocole_d'accord	0.6938
3	l'accord	0.6724
4	signer	0.6587
5	accord-cadre	0.6227
6	arrangement	0.6008
7	parapher	0.5956
8	ratifier	0.5915
9	compromis	0.5790
10	négocier	0.5766

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	inefficace	0.7565
2	insuffisant	0.6909
3	inutile	0.6674
4	contre-productif	0.6548
5	inadapté	0.6486
6	injuste	0.6425
7	incohérent	0.6389
8	inopérant	0.6205
9	absurde	0.6190
10	inacceptable	0.6146

TABLE B.37: 10 most similar terms to the 10 centroids of the cluster model for the French lexicon for Efficacy Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.3.4 Centroids of the Lexicon for Efficiency Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	blocage	0.6857
2	impasse	0.6769
3	paralysie	0.6349
4	paralyser	0.5900
5	échec	0.5823
6	désaccord	0.5813
7	compromis	0.5716
8	opposition	0.5684
9	négociation	0.5634
10	enliser	0.5574

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	car	0.6753
2	enfin	0.6671
3	surtout	0.6658
4	peut-être	0.6587
5	parce_que	0.6483
6	vraiment	0.6452
7	donc	0.6409
8	problème	0.6393
9	penser	0.6369
10	savoir	0.6365

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	insuffisance	0.7425
2	dysfonctionnement	0.7095
3	carence	0.7038
4	inefficacité	0.6937
5	incohérence	0.6590
6	incompétence	0.6462
7	manque	0.6451
8	incapacité	0.6385
9	impuissance	0.6340
10	déficience	0.6294

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	déficit	0.7763
2	déficit_budgétaire	0.7603
3	dette	0.7127
4	dépense	0.7075
5	endettement	0.6940
6	recette_fiscal	0.6915
7	budgétaire	0.6760
8	PIB	0.6401
9	budget	0.6372
10	rentrée_fiscal	0.6254

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	objectif	0.6917
2	engagement	0.6776
3	proposition	0.6353
4	effort	0.6192
5	mesure	0.6063
6	plan	0.6015
7	réforme	0.5778
8	programme	0.5698
9	ambitieux	0.5695
10	initiative	0.5553

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	développement	0.6448
2	efficacité	0.6237
3	environnemental	0.6022
4	écologique	0.5966
5	économique	0.5860
6	ressource	0.5798
7	énergétique	0.5716
8	industriel	0.5490
9	innovation	0.5331
10	efficacité_énergétique	0.5303

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	baisse	0.7741
2	hausse	0.7604
3	augmenter	0.7491
4	baisser	0.7056
5	augmentation	0.6522
6	diminuer	0.6517
7	progresser	0.6449
8	progression	0.6352
9	en_hausse	0.6252
10	en_baisse	0.6174

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	coûteux	0.7795
2	onéreux	0.6914
3	efficace	0.6732
4	moins_coûteux	0.6712
5	inefficace	0.6290
6	performant	0.6137
7	efficient	0.6007
8	centralisation_bureaucratique	0.6004
9	compliquer	0.5951
10	complexe	0.5853

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	attendre	0.7104
2	demander	0.7074
3	donc	0.6725
4	poursuivre	0.6690
5	enfin	0.6659
6	alors_que	0.6624
7	décider	0.6623
8	accepter	0.6572
9	par_ailleurs	0.6530
10	prévoir	0.6528

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	accélérer	0.6687
2	aboutir	0.5965
3	rapide	0.5811
4	retarder	0.5805
5	ralentir	0.5370
6	accélération	0.5320
7	lent	0.5319
8	enclencher	0.5259
9	processus	0.5202
10	concrétiser	0.5193

TABLE B.38: 10 most similar terms to the 10 centroids of the cluster model for the French lexicon for Efficiency Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.3.5 Centroids of the Lexicon for Epistemic Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	analyse	0.7281
2	rapport	0.6836
3	expertise	0.6728
4	évaluation	0.6711
5	conclusion	0.6684
6	expert	0.6653
7	diagnostic	0.6271
8	étude	0.5868
9	synthèse_épurer	0.5832
10	méthodologie	0.5749

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	scientifique	0.8062
2	chercheur	0.7276
3	biologiste	0.6606
4	climatologue	0.6236
5	science	0.5989
6	jean-paul_Watteau	0.5793
7	climatologie_australien	0.5785
8	physicien	0.5711
9	technologique_OPECST	0.5644
10	expert	0.5620

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	souligner	0.8152
2	préciser	0.7971
3	indiquer	0.7966
4	affirmer	0.7903
5	expliquer	0.7854
6	estimer	0.7769
7	assurer	0.7554
8	ajouter	0.7527
9	rappeler	0.7356
10	déclarer	0.7340

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	université	0.8479
2	professeur	0.8122
3	l'université	0.7523
4	christian_Fornari	0.7474
5	l'Université	0.7423
6	Sorbonne	0.7263
7	emmanuel_Fraisse	0.7260
8	Université	0.7242
9	Paris_I-Sorbonne	0.7173
10	Harvard	0.7100

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	expert_intergouvernemental	0.7784
2	climat_IPCC	0.7651
3	climat_Giec	0.7647
4	climat_GIEC	0.7531
5	Giec	0.7391
6	intergouvernemental	0.7248
7	IPCC	0.7024
8	GIEC	0.6838
9	panel_intergouvernemental	0.6635
10	http://www.ipcc.ch	0.6105

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	climat	0.7623
2	environnement	0.7247
3	réchauffement_climatique	0.6691
4	changement_climatique	0.6660
5	réchauffement	0.6633
6	économie	0.5974
7	climatique	0.5770
8	changement	0.5730
9	biodiversité	0.5556
10	planète	0.5097

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	Research	0.7209
2	Environment	0.7132
3	and	0.7068
4	Institute	0.7053
5	Toward	0.6968
6	Role	0.6778
7	of	0.6741
8	evidence	0.6550
9	Overview	0.6492
10	of_our	0.6490

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	sociologue	0.7633
2	politologue	0.7319
3	spécialiste	0.7297
4	économiste	0.7211
5	géographe	0.6870
6	historien	0.6433
7	chercheur	0.6396
8	démographe	0.6212
9	consultant	0.6103
10	analyste	0.6091

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	université_Paris-I-Sorbonn	0.7668
2	science	0.7493
3	christian_Fornari	0.7172
4	Jacquillat_professeur	0.7168
5	alain_Berthoz	0.7151
6	Kepel_professeur	0.7110
7	Genève_avocatet	0.7066
8	Paris-Sud-XI	0.7022
9	www.finances-europe.com	0.6986
10	Paris_I-Sorbonne	0.6983

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	recherche	0.7357
2	étude	0.6873
3	développement	0.6384
4	publicitaire_Irep	0.6265
5	domaine	0.5901
6	spécialiser	0.5881
7	programme	0.5723
8	statistiqueà	0.5450
9	NHGRI	0.5445
10	EPF_Eawag	0.5427

TABLE B.39: 10 most similar terms to the 10 centroids of the cluster model for the French lexicon for Epistemic Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.3.6 Centroids of the Lexicon for Legality Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	constitutionnel	0.7203
2	constitution	0.6813
3	loi	0.6802
4	présentement_examiner	0.6320
5	loi_organique	0.6166
6	texte	0.5915
7	révision_constitutionnel	0.5830
8	conseil_constitutionnel	0.5827
9	88-6	0.5719
10	50-1	0.5684

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	procès	0.7868
2	accusé	0.7538
3	acquittement	0.7301
4	juge	0.7130
5	prévenu	0.6891
6	procureur	0.6796
7	tribunal	0.6742
8	avocat	0.6737
9	condamnation	0.6726
10	magistrat	0.6709

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	corruption	0.7705
2	soupçon	0.7362
3	malversation	0.7312
4	soupçonner	0.6736
5	favoritisme	0.6487
6	accusation	0.6462
7	pot-de-vin	0.6400
8	irrégularité	0.6311
9	présumer	0.6292
10	fraude	0.6288

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	escroquerie	0.7744
2	fraude_fiscal	0.7479
3	blanchiment	0.7469
4	recel	0.7325
5	escroquerie_recel	0.7110
6	abus	0.7045
7	détournement	0.6971
8	corruption_passif	0.6940
9	complicité	0.6889
10	blanchiment_aggraver	0.6859

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	respecter	0.7214
2	enfreindre	0.7208
3	contrevenir	0.7115
4	violer	0.6908
5	règle	0.6559
6	toute_entors	0.6476
7	violation	0.6451
8	contraire	0.6305
9	respect	0.6304
10	conforme	0.6170

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	délit	0.7514
2	pénal	0.6794
3	infraction	0.6674
4	corruption	0.6553
5	blanchiment	0.6300
6	crime	0.6283
7	fraude_fiscal	0.6187
8	criminel	0.6087
9	fraude	0.5937
10	racket_escroquerie	0.5862

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	enfin	0.5697
2	injustices	0.5603
3	sentiment_d'injustice	0.5552
4	injustice	0.5474
5	donc	0.5424
6	vouloir	0.5411
7	sentiment	0.5403
8	car	0.5383
9	surtout	0.5292
10	aujourd'hui	0.5285

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	illégal	0.7848
2	interdire	0.7375
3	illicite	0.7248
4	prohiber	0.7050
5	autoriser	0.6585
6	licite	0.6488
7	illégalité	0.6285
8	interdiction	0.6262
9	légal	0.6223
10	légalement	0.6118

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	incarcérer	0.8156
2	inculper	0.7449
3	écrouer	0.7346
4	condamner	0.7243
5	prison	0.7053
6	meurtre	0.7051
7	détention_provisoire	0.7044
8	incarcération	0.6834
9	détention	0.6611
10	comparaître	0.6600

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	cour	0.7969
2	tribunal	0.7604
3	cassation	0.7318
4	justice	0.7238
5	cour_suprême	0.6998
6	Justice_CEJ	0.6904
7	jugement	0.6859
8	tribunal_administratif	0.6746
9	appel	0.6732
10	appel-nullité	0.6725

TABLE B.40: 10 most similar terms to the 10 centroids of the cluster model for the French lexicon for Legality Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.3.7 Centroids of the Lexicon for Participation Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	élire	0.7877
2	élection	0.7830
3	réélire	0.7155
4	réélection	0.7023
5	scrutin	0.6704
6	élu	0.6653
7	élection_municipal	0.6594
8	réélu	0.6488
9	élus	0.6432
10	municipal	0.6390

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	appeler	0.7791
2	réclamer	0.7422
3	demander	0.7342
4	soutenir	0.7328
5	décider	0.7141
6	annoncer	0.6501
7	opposer	0.6465
8	refuser	0.6463
9	appel	0.6418
10	accepter	0.6399

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	coconstruction	0.6777
2	participatif	0.6705
3	concertation	0.6190
4	indépendant_multipartit	0.5932
5	configuration_multifactoriel	0.5834
6	codécision	0.5453
7	agenda_2063	0.5151
8	collaboratif	0.4947
9	démocratie_participatif	0.4795
10	citoyenne	0.4677

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	vote	0.7778
2	électeur	0.7735
3	votant	0.7601
4	suffrage	0.7423
5	voter	0.7070
6	scrutin	0.6936
7	abstention	0.6696
8	suffrage_exprimer	0.6564
9	voix	0.6540
10	majorité	0.6506

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	habitant	0.7663
2	population	0.6783
3	ville	0.6520
4	gens	0.6337
5	communauté	0.5632
6	d'habitants	0.5592
7	région	0.5580
8	citoyen	0.5437
9	l'habitant	0.5400
10	personne	0.5328

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	initiative	0.7230
2	proposition	0.6859
3	propos	0.6283
4	engagement	0.6178
5	parole	0.6159
6	soutien	0.6149
7	déclaration	0.6149
8	décision	0.6009
9	texte	0.5850
10	avis	0.5629

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	participer	0.7090
2	organiser	0.6825
3	inviter	0.6799
4	consacrer	0.6708
5	public	0.6141
6	partager	0.5524
7	tenir	0.5523
8	poursuivre	0.5330
9	présent	0.5283
10	lors_de	0.5204

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	militant	0.7128
2	mouvement	0.6575
3	mobiliser	0.6425
4	rassembler	0.5708
5	rejoindre	0.5699
6	manifestation	0.5688
7	manifester	0.5451
8	soutenir	0.5406
9	élue	0.5360
10	sympathisant	0.5282

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	votation	0.7945
2	référendum	0.7495
3	initiative_populaire	0.7468
4	consultation_populaire	0.6682
5	UDC	0.6293
6	votation_populaire	0.6131
7	acceptation	0.5996
8	consultation	0.5888
9	article_constitutionnel	0.5426
10	antiminarets	0.5294

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	démocratie	0.7736
2	citoyen	0.7476
3	démocratique	0.7240
4	peuple	0.7116
5	politique	0.5952
6	civique	0.5918
7	démocratie_direct	0.5764
8	concitoyen	0.5687
9	aspiration	0.5672
10	démocratie_participatif	0.5591

TABLE B.41: 10 most similar terms to the 10 centroids of the cluster model for the French lexicon for Participation Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.3.8 Centroids of the Lexicon for Representation Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	parti	0.6087
2	communiste	0.5985
3	parti_communiste	0.5824
4	UDC	0.5799
5	parti_radical-démocratique	0.5659
6	Al-Massar	0.5612
7	PDC	0.5595
8	sceptique_pdc	0.5578
9	parti_radical	0.5576
10	HZDS	0.5519

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	candidat	0.7068
2	élire	0.6896
3	socialiste	0.6464
4	maire	0.6428
5	PS	0.6271
6	président	0.6119
7	UMP	0.6076
8	mairie	0.6052
9	municipal	0.5991
10	présidentiel	0.5908

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	assemblée_national	0.8430
2	sénat	0.8261
3	assemblée	0.8136
4	parlementaire	0.7309
5	l'Assemblée_national	0.7168
6	l'Assemblée	0.7021
7	Assemblée	0.6543
8	CMP_7	0.6421
9	mixte_paritaire	0.6326
10	Assemblée-Sénat	0.6244

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	CDU	0.8128
2	SPD	0.7742
3	parti_social-démocrate	0.7556
4	chancelier	0.7513
5	social-démocrate	0.7498
6	Union_chrétienne-démocrate	0.7239
7	ministre-présidente	0.6816
8	chancelier_chrétienne-démocrate	0.6717
9	CSU	0.6473
10	ministre-président	0.6438

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	PS	0.8068
2	UDI	0.7976
3	UDF	0.7938
4	UMP	0.7897
5	socialiste	0.7869
6	modem	0.7798
7	centriste	0.7608
8	député	0.7424
9	PRG	0.7411
10	PCF	0.7222

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	parlement	0.6461
2	Parlement	0.6155
3	colégislateur_sans_doute	0.6005
4	Econ_ainsi_que	0.5787
5	moscovici_chahuter	0.5732
6	Com-mission	0.5627
7	assemblée	0.5589
8	CMP_Assemblée-Sénat	0.5589
9	siègerait	0.5516
10	good_cops	0.5468

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	responsable	0.7154
2	charger	0.6750
3	par_ailleurs	0.6543
4	affirmer	0.6538
5	déclarer	0.6488
6	indiquer	0.6480
7	membre	0.6477
8	conseil	0.6457
9	représentant	0.6439
10	assurer	0.6424

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	parti	0.8246
2	gauche	0.8107
3	droite	0.8076
4	parti_socialiste	0.7061
5	opposition	0.6929
6	majorité	0.6885
7	socialiste	0.6868
8	centriste	0.6831
9	PS	0.6573
10	gouvernement	0.6478

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	ministre	0.7950
2	ex-ministre	0.7304
3	vice-ministre	0.6802
4	jim_Flaherty	0.6730
5	vice-premier_ministre	0.6721
6	affaire_étranger	0.6554
7	bruno_Archi	0.6532
8	elena_Salgado	0.6463
9	Premier_ministre	0.6339
10	Ahmad_Vahidi	0.6307

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	législateur	0.7582
2	législation	0.7041
3	constitutionnel	0.6650
4	loi	0.6371
5	présentement_examiner	0.6276
6	décision-cadre_bien_que	0.6246
7	réglementation	0.6033
8	légiférer	0.5999
9	cadre_légal	0.5922
10	constitution	0.5900

TABLE B.42: 10 most similar terms to the 10 centroids of the cluster model for the French lexicon for Representation Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.3.9 Centroids of the Lexicon for Stakeholder Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	partie_prenant	0.6403
2	partenaire	0.6065
3	collectivité	0.6061
4	Etat	0.5695
5	collectivité_local	0.5454
6	créancier	0.5343
7	concessionnaire_garagiste	0.5335
8	concertation	0.5213
9	Etats	0.5188
10	négociation	0.4981

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	organisation	0.6986
2	Homme_Amnesty	0.6304
3	fédération	0.6125
4	carrosserie_FFC	0.6021
5	association	0.5899
6	tourisme_Effat	0.5783
7	tourisme_EFFAT	0.5718
8	TASZ	0.5615
9	ONG	0.5599
10	Al-Haq	0.5512

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	manifester	0.7176
2	manifestation	0.6806
3	soutenir	0.6641
4	dénoncer	0.6548
5	réclamer	0.6530
6	appeler	0.6438
7	défendre	0.6357
8	organiser	0.6136
9	mobiliser	0.6092
10	exprimer	0.6050

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	responsable	0.7121
2	représentant	0.6531
3	vice-président	0.6457
4	membre	0.6433
5	affirmer	0.6381
6	directeur	0.6329
7	souligner	0.6326
8	porte-parole	0.6314
9	par_ailleurs	0.6281
10	assurer	0.6263

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	Climat-France	0.7538
2	WWF	0.7043
3	Greenpeace	0.6922
4	écologiste	0.6764
5	ONG	0.6682
6	Greenpeace_Attac	0.6545
7	Climat_RAC	0.6541
8	Greenpeace_WWF	0.6424
9	Christophe_Aubel	0.6114
10	association	0.6101

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	expliquer	0.7372
2	affirmer	0.7362
3	ajouter	0.7262
4	assurer	0.7236
5	ainsi	0.7205
6	estimer	0.7146
7	souligner	0.7030
8	donc	0.7018
9	déclarer	0.6972
10	également	0.6886

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	économiste	0.6903
2	banque	0.6164
3	analyste	0.5789
4	financier	0.5447
5	bancaire	0.5382
6	secteur_bancaire	0.5290
7	monétaire	0.5273
8	expert	0.5224
9	analyse	0.5217
10	Scott_Bugie	0.5216

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	maire	0.6801
2	ville	0.6798
3	commune	0.6769
4	mairie	0.6744
5	région	0.6560
6	agglomération	0.6395
7	municipalité	0.6147
8	Talange	0.5626
9	communauté_urbain	0.5497
10	UMP	0.5491

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	ONG	0.7386
2	Amnesty_international	0.7281
3	Human_Rights	0.6703
4	non_gouvernemental	0.6486
5	Amnesty	0.6457
6	Watch	0.6347
7	HRW	0.6316
8	Defenders	0.6136
9	Human_Right	0.6117
10	www.hrw.org	0.6110

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	lobby	0.7415
2	industrie	0.6157
3	lobbying	0.6083
4	lobbyiste	0.6026
5	industriel	0.5872
6	puissant_lobby	0.5654
7	industrie_chimique	0.5638
8	l'industrie	0.5310
9	industrie_pharmaceutique	0.5299
10	secteur	0.5290

TABLE B.43: 10 most similar terms to the 10 centroids of the cluster model for the French lexicon for Stakeholder Frames in the semantic space of the word2vec model, ordered by cosine similarity

B.2.3.10 Centroids of the Lexicon for Transparency Frames

10 most similar entries to centroid 1		
Rank	Word	Similarity
1	informer	0.8084
2	alerter	0.7409
3	avertir	0.6946
4	renseigner	0.6761
5	rendre_compte	0.6711
6	prévenir	0.5857
7	signaler	0.5611
8	consulter	0.5390
9	inquiéter	0.5068
10	vérifier	0.4973

10 most similar entries to centroid 2		
Rank	Word	Similarity
1	transparent	0.7018
2	crédible	0.6942
3	fiable	0.6609
4	efficace	0.6495
5	pertinent	0.6402
6	honnête	0.6365
7	équitable	0.6255
8	précis	0.6102
9	rigoureux	0.6101
10	cohérent	0.6062

10 most similar entries to centroid 3		
Rank	Word	Similarity
1	mensonge	0.7723
2	mentir	0.6457
3	irresponsable	0.6250
4	désinformation	0.6129
5	menteur	0.5995
6	mauvais_foi	0.5822
7	vérité	0.5684
8	calomnie	0.5645
9	contre-vérité	0.5601
10	scandaleux	0.5577

10 most similar entries to centroid 4		
Rank	Word	Similarity
1	manifestation	0.7981
2	message	0.7981
3	rassemblement	0.5813
4	manifestant	0.5572
5	mobilisation	0.5488
6	protestation	0.5418
7	manifester	0.5409
8	mouvement	0.5095
9	discours	0.5017
10	journée	0.4960

10 most similar entries to centroid 5		
Rank	Word	Similarity
1	affirmer	0.8279
2	indiquer	0.8106
3	préciser	0.8018
4	confirmer	0.8010
5	expliquer	0.7903
6	déclarer	0.7551
7	souligner	0.7550
8	ajouter	0.7421
9	révéler	0.7329
10	rappeler	0.7292

10 most similar entries to centroid 6		
Rank	Word	Similarity
1	information	0.7173
2	document	0.7041
3	divulgation	0.6502
4	divulguer	0.6256
5	secret	0.6187
6	confidentiel	0.6175
7	révélation	0.6087
8	WikiLeaks.Le	0.5893
9	d'Edward_Snowden	0.5788
10	rapport	0.5545

10 most similar entries to centroid 7		
Rank	Word	Similarity
1	transparence	0.7179
2	honnêteté	0.7150
3	impartialité	0.6843
4	clarté	0.6830
5	sincérité	0.6827
6	crédibilité	0.6754
7	intégrité	0.6623
8	cohérence	0.6471
9	équité	0.6400
10	fiabilité	0.6399

10 most similar entries to centroid 8		
Rank	Word	Similarity
1	dissimuler	0.7365
2	cacher	0.6063
3	camoufler	0.5929
4	dissimulation	0.5873
5	obscur	0.5691
6	opaque	0.5638
7	masquer	0.5564
8	occulte	0.5229
9	blanchiment	0.5151
10	frauduleux	0.4941

10 most similar entries to centroid 9		
Rank	Word	Similarity
1	rendre_transparents	0.6329
2	information-communication	0.5714
3	publications	0.5403
4	application.bruxelle	0.5008
5	transparentes	0.4965
6	claire	0.4792
7	transparence_claironner	0.4727
8	PROPOSITION_DE	0.4560
9	luxembourg_tranche	0.4449
10	approprié	0.4431

10 most similar entries to centroid 10		
Rank	Word	Similarity
1	bien-fondé	0.7735
2	pertinence	0.7703
3	véracité	0.6521
4	contestable	0.6044
5	utilité	0.6006
6	validité	0.5455
7	légitimité	0.5160
8	contester	0.5035
9	saboter	0.5033
10	légalité	0.5014

TABLE B.44: 10 most similar terms to the 10 centroids of the cluster model for the French lexicon for Transparency Frames in the semantic space of the word2vec model, ordered by cosine similarity

C

Codebook for Framing Analysis

C.1 Original Codebook for the Framing Analysis

We include the codebook for the framing analysis into the appendix in order to provide with in-detail information concerning the coded frames.

Note that this is a copy of the text in the codebook and has not been reformulated in any way due to preserve the original state of the codebook. The codebook has been written by Bruno Wueest.

[start of codebook]

Guidelines for the Frame Analysis in Module 1 bw / 24.3.2016

Goal We collect data on the democratic legitimacy of new forms of governance as it is reported in newspapers. Democratic legitimacy frames can relate to the proper working of democratic mechanisms, the participation of the potentially affected citizens, the representation of the citizen's interests, and the authorities' diligent work as well as effectiveness.

PLEASE NOTE: There are always questions and insecurities during an annotation. You have the possibility to include notes for every annotation during the coding process.

However, please ask Bruno Wueest (wueest@ipz.uzh.ch) in all cases for which the level of insecurity is especially high. It is better to ask questions than to put the data collection into danger with unassertive annotations!

Unit of observation Our unit of observation is a reference to a particular kind of democratic legitimacy. Hence, we consider all text passages related to democratic legitimacy as relevant, independent from the question whether this reference is connected to an entity we are interested in or not. For the automated analysis, it is of utmost importance to annotate all references to democratic legitimacy, in order to avoid false negatives.

General remarks

Each text has to be read twice:

1. During the first reading, all text fragments relevant to in- or output legitimacy are identified and highlighted. This first identification of relevant text passages is crucial for all further steps of the annotation, so please follow the following definition as close as possible: A text passage is considered relevant if there is talk about democratic legitimacy, which is explicitly linked to an actor or act of governance, i.e. policy making or politics in general terms. Hence, we do not annotate pure descriptions of emotional states, aesthetic properties, personal behavior or individual attitudes, unless they are explicitly linked to governance. As for the text fragment to be highlighted, we take the minimal number of words that together refer to a legitimacy frame. Please consider the following example of an efficiency frame: "**Wirtschaftsförderung** betreibt zum Beispiel die Organisation Greater Zurich Area **viel effizienter**." Here, the words 'Wirtschaftsföderung' and 'viel effizienter' should be marked.

PLEASE NOTE: There is no restriction of the number of text fragments for each text. Every document can thus have none, one or multiple legitimacy frames.

PLEASE NOTE: It is important that you start with the identification of frames and not the entities, since the frames are the key indicators of the analysis.

PLEASE NOTE: Documents are presented by language. Further, every document name has an increment as prefix. You can use this increment to orient yourself at the beginning of a session. In other words, please note at which document you stopped at the end of a session so you know where to start at the beginning of the next session.

2. During the second reading, the labels of the two indicators substance and evaluation (see below) have to be added to the relevant text fragments.

3. During a third reading, frames and target entities should be linked to each other. Hence, it has to be indicated if an explicit link between the legitimacy frames and the occurrence of a governance entity can be made.

4. In a last step, you have to link several occurrences of the same target or suggested entities as synonyms.

PLEASE NOTE: A text passage is considered relevant if there is an explicit link to an actor or act of governance, i.e. policy making or politics in general terms. Hence, we do not annotate pure descriptions of emotional states, aesthetic properties, personal behavior or individual attitudes, unless they are explicitly linked to governance. Sometimes it will be difficult to separate a personal behavior from a political act, if such personal behavior comes from someone who is a political actor. For example, if a minister were to say something about representing a certain constituency, this is linked to governance. If the same minister is portrait as having a passion for football, we do not consider this relevant. Also, sometimes, media will talk about fictitious, imagined actors or acts of governance, e.g. if someone suggests a new agency to regulate a specific policy field. We consider such references relevant as well.

PLEASE NOTE: In general, we take the perspective of the media when deciding on indicators and labels. However, we annotate all instances of democratic legitimacy. Hence, direct speech or paraphrases of statements by non-media actors is considered relevant as well, even if the different instances of democratic legitimacy in a document are contradictory.

PLEASE NOTE: Try to use as few contextual knowledge, which is not explicitly mentioned in a document, as possible. Not all coders have the same contextual knowledge, a direct application of this knowledge thus potentially leads to coder bias.

Indicators

Three indicators need to be annotated for every instance of democratic legitimacy. The first refers to the substance of a frame, the second to the evaluation of a frame, and the third to a possible relationship of the frame to the governance entities under concern. In addition, different occurrences of the same entity have to annotated with a synonym relation.

Frame substance

In the context of this analysis, we understand frames as schemata of interpretation that refer to the source of legitimacy of governance entities. In this analysis, we separate input- from throughput- and output-oriented frames. Input legitimacy is present if aspects related to participation, deliberation or representation are mentioned. Throughput

legitimacy occurs if frames refer to the transparency, legality or accountability of governance. Output legitimacy, on the other hand, is given if text fragments refer to the efficiency, effectiveness, and the empowerment of governance are mentioned.

Here is an example of two input legitimacy frames: " 'I am certain we will have the political intelligence to assure a **balanced representation of each of our countries and peoples**, with full respect for the **judicial equality of states**,' Lula said." More precisely, we assign one of the following values to each text fragment found relevant

Input legitimacy

I_1	Representation	The power to decide rests with the people/citizens. The interests of the people/citizens are represented. A broad range of interests is included into the opinion formation and decision making processes. Elected representatives are involved in the opinion formation and decision-making processes. Elected representatives or representative institutions like parliaments or other bodies represent the interests of the people/citizens.
I_2	Participation	Citizens are involved in the opinion formation and decision making processes.
I_3	Deliberation	Political decisions are reached by careful consideration and/or discussion among the stakeholders involved (pluralist decision making). Decision-makers are able to change their positions in the light of the better arguments.
I_4	Epistemic	Experts (scientists, specialists etc.) shall be involved in the opinion formation and decision-making processes.
I_5	Stakeholder	Civil society and business actors are involved in the opinion formation and decision making processes.

Throughput legitimacy

T_1	Transparency	Policy processes, decisions and outcomes are transparent to the public. Information about decision-making processes and decisions is accessible and/or actively disseminated to the people/citizens.
T_2	Accountability	Policy processes, decisions and outcomes are controllable or, if necessary, corrigible by the public, elected representatives or civil society actors.
T_3	Legality	National and/or international laws are respected. Decision-makers do not overstep their competences.

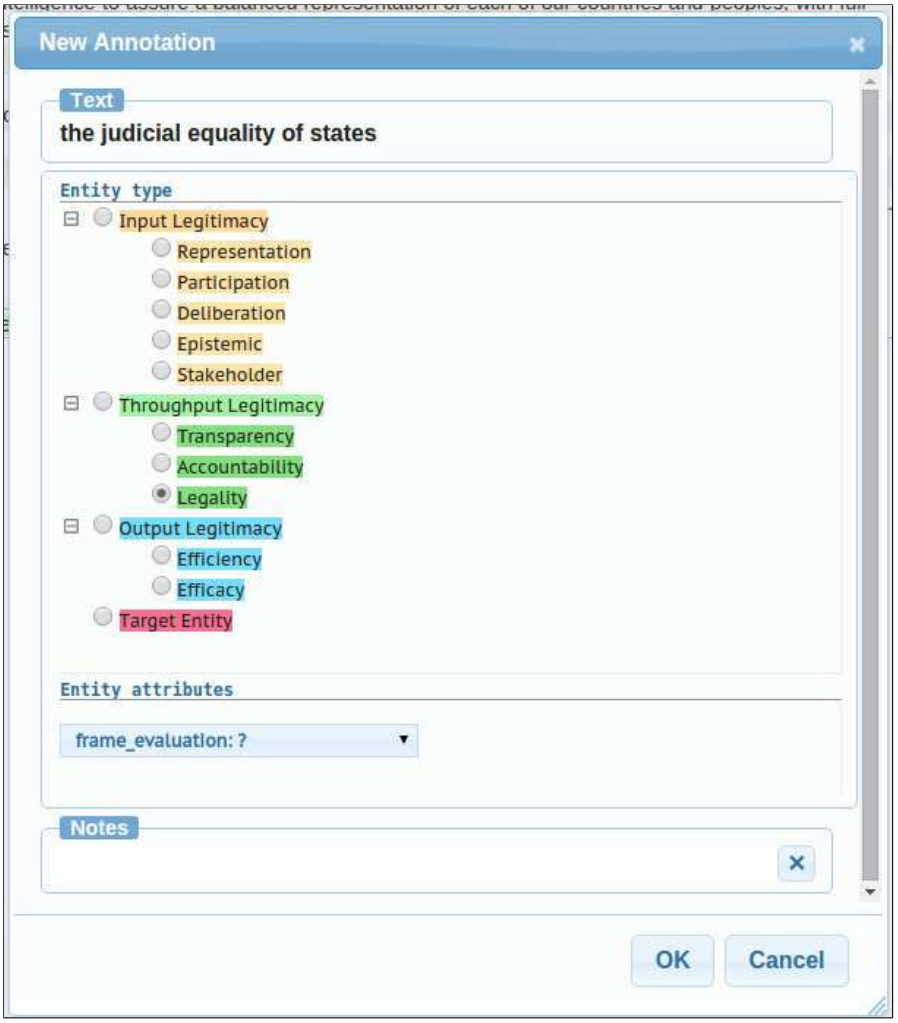
Output legitimacy

O_1	Efficiency	Political decisions were reached efficiently. Measures are cost-efficient.
O_2	Efficacy	Measures are successful, i.e. they lead to actual changes in the regulated policy area. Also, measures are sufficient to solve the problem at stake. Such a solution might refer to the common good, productivity, less externalities, better distributive justice, human rights etc. in the regulated policy areas.

PLEASE NOTE: We annotate all occurrences of these types of legitimacy in disregard of their direction. Hence, we also record negations (e.g. 'no elected representatives are involved' = Representation) and subjunctives (some type of legitimacy should or could be present).

In the interface, we have just to highlight a text passage, then a dialogue will pop up where we check the corresponding category of frame substance. Additional text

fragments can be added with the "Add frag." function (at the bottom left, not shown in the picture).



PLEASE NOTE: If a text passage refers to more than one frame, it can be highlighted and labeled as many times as necessary.

PLEASE NOTE: We have introduced a general category labeled 'Democratic Legitimacy'. This label should only be assigned for text passages where it is clear that it is about democratic legitimacy, but it is impossible to separate which kind.

PLEASE NOTE: If you do not find any references to democratic legitimacy in a text, please indicate this as follows: Highlight any word, preferably at the beginning of the document. Don't make any annotations except a note in the comment entry field that says: "No reference".

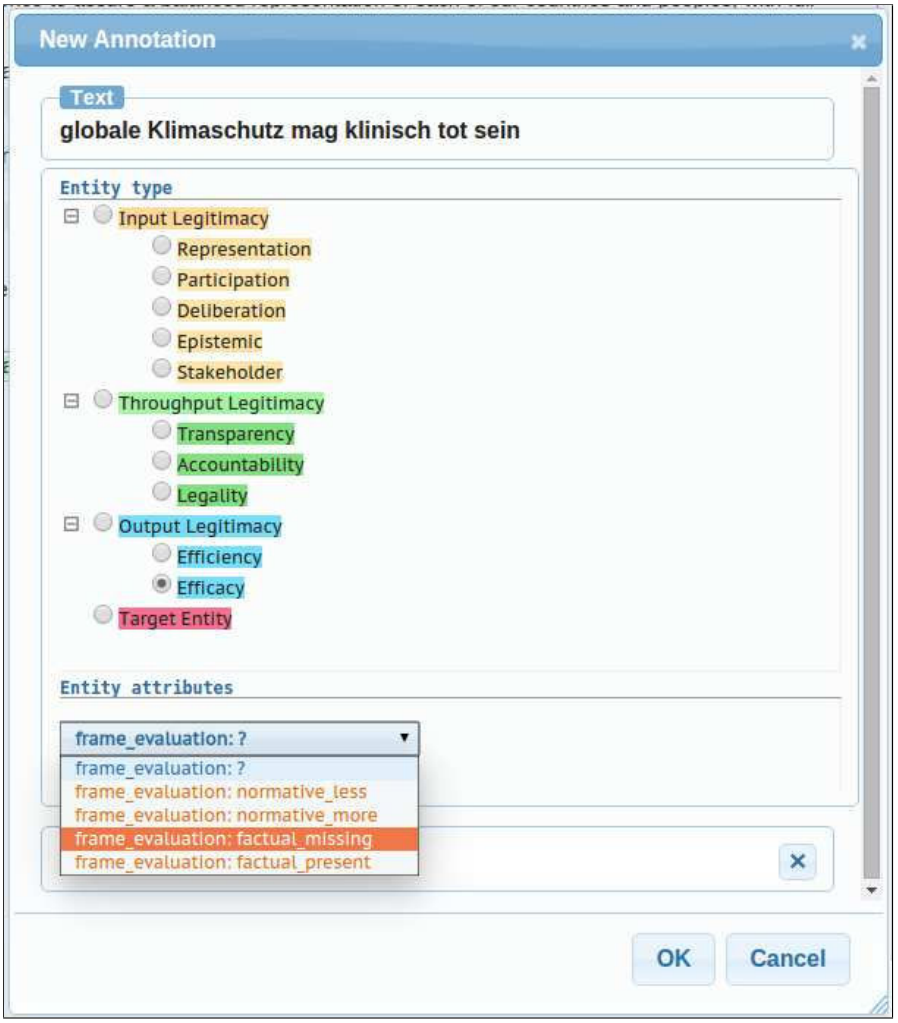
Frame evaluation

The second indicator refers to the evaluation of frames. This means that we assess the evaluation of the kind of legitimacy annotated as the substance of frames. In general, we separate factual evaluations (a certain kind of legitimacy is present or missing) from normative evaluations (legitimacy should be enhanced or attenuated). more precisely, this indicator has the following values:

F_1	present	It is presented as a fact that legitimacy is given.
F_2	missing	It is presented as a fact that that legitimacy is lacking.
N_1	more	It is claimed that legitimacy should be increased.
N_2	less	It is claimed that legitimacy should be decreased.

In the following example, it is presented as a fact that the efficacy of the Kypoty protocol is completely missing: "Der **globale Klimaschutz mag klinisch tot sein**, aber der **Kyoto**-Prozess läuft wie ein Zombie einfach weiter"

In the interface, the categories of frame evaluation are included in the same panel we have to right-click on the highlighted text fragment, and then check the corresponding category.



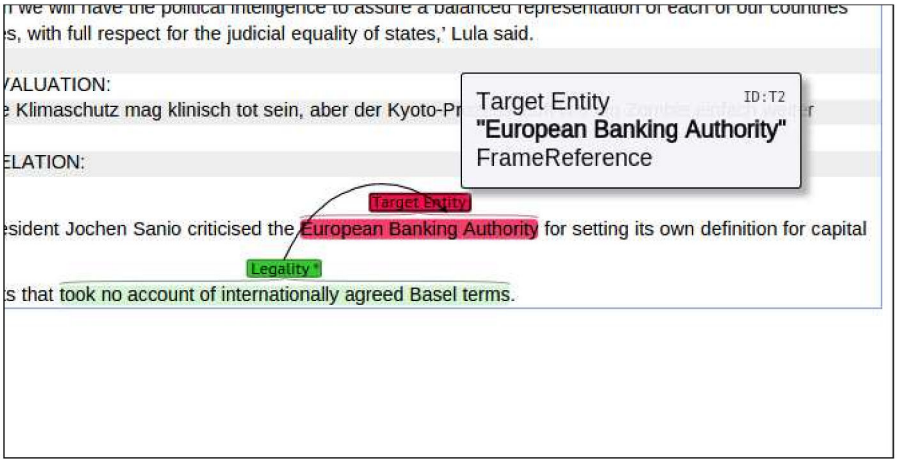
PLEASE NOTE: The distinction between normative and factual can be difficult. Sometimes, the language used in a text passage can provide clear hints for one or the 7other category. E.g., certain modal verbs (shall, must) or subjunctives (could, would) are clear signs for a normative evaluation.

Frame relation In the next step, we are interested in the link between the text fragments on legitimacy frames and the governance entities occurring in a text. Hence, we establish a relationship whether a direct reference between the frame and the entity is established in the document. Please consider the following example:

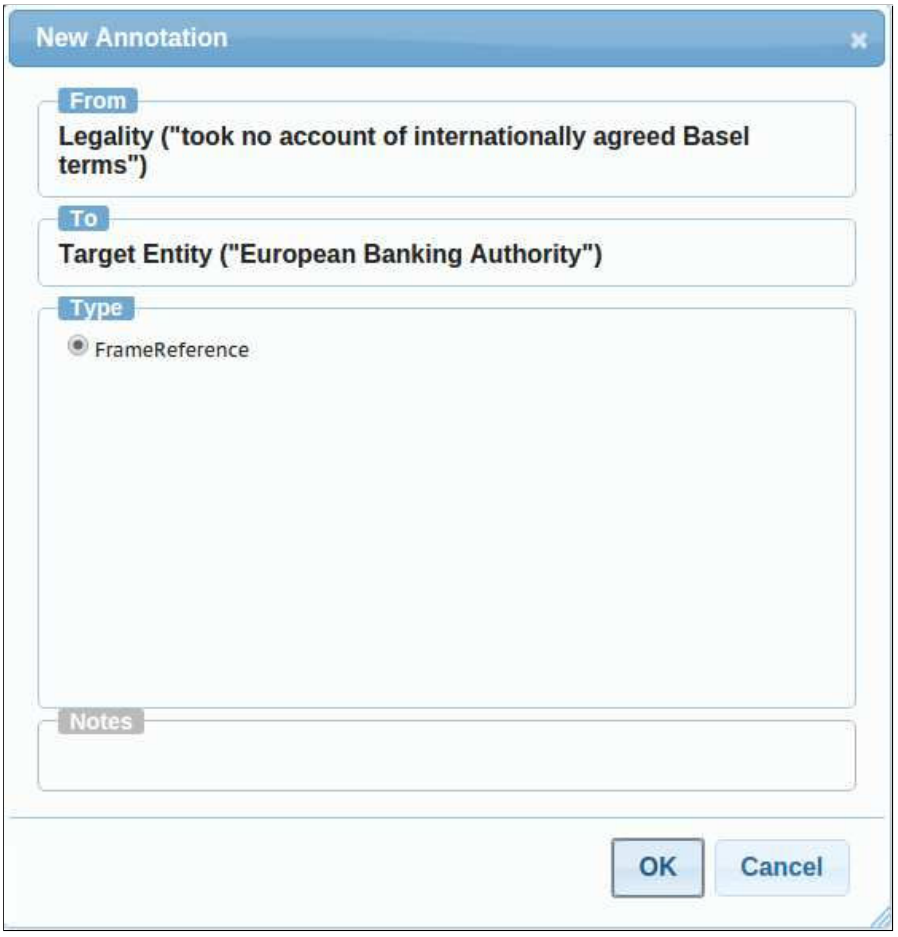
"BaFin's president Jochen Sanio criticised the **European Banking Authority** for setting its own definition for capital benchmarks that took no account of **internationally agreed Basel terms**."

In this example, the legality frame "internationally agreed Basel terms" needs to be linked to the entity 'European Banking Authority'. For the entities that are highlighted in the text, all possible relations have to be indicated. For other entities you discover, but which are not highlighted, these relations should be indicated as well, but only if the relation is quickly identifiable. The defining criterium for actors (organizations and individuals) or agreements (treaties, protocols etc.) is that they can be linked to any acts of governance. Hence, there are entities such as politicians that are linked to governance by default. Other potential entities, for example football players, only are considered relevant target entities if they are explicitly linked to an act of governance in the text. In cases of doubt it is better to annotate a target entity than to leave it aside.

In the interface, an arrow needs to be drawn from the frame to the entity it refers to. This can be done by keeping the mouse clicked and moving from the frame to the entity. Please note that this will only work in the direction frame → target entity.



Subsequently, in a pop-up window you have to check the frame relation. Since that is the only reference we annotate there won't be a 'choice' but you have to confirm it, and you can make a note to this annotation.



PLEASE NOTE: We only consider references to entities which are predefined by the projects as candidates for a relation. See the initial table of this codebook.

PLEASE NOTE: It does not matter how many words and sentences are placed between a frame and an entity. If there is a relationship, it does not matter how large the span is.

Synonym relation

In a last step, different occurrences of the same entity should be annotated. For example, the 'Kyoto protocol', 'Kyoto negotiations', 'Kyoto agreement' etc. all are referring to the same entity and should therefore be linked.

PLEASE NOTE: We are annotating linguistically synonymous entities. Different entities that belong to the same concept are not linked to each other. E.g., if the

'Kyoto protocol', the UN as hosting authority of the negotiations and a participant of the negotiations all are occurring in a text, they should not be linked to each other.

PLEASE NOTE: You do not have to annotate all synonyms of an entity in a text as target entities, i.e. you don't have to look for additional target entities at this point. It is sufficient if you link words you already annotated as target entities or suggested entities with each other.

In the interface, you can simply draw an arrow from one target entities to the other to indicate synonyms. It will automatically be labelled as an 'isSynonym' relation. You can link as many entities to each other as you want. However, you do not have to link every synonym with every other synonym, it is sufficient if they are all chained to each other. E.g. if you have the words A, B, and C in a text, which refer to the same entity

Manual

The texts will be annoted in a customised, web-based user interface on the basis of brat (see http://brat.nlplab.org/) including the already highlighted occurrence of the governance entities as identified by the salience analysis. You can access the interface from any computer with any browser at any time.

A. First, you need to access and login to the annotation interface:

1. Point a browser window to the link that corresponds to your project.

2. You will be directed to the overview of the collection of documents that is assigned to you for annotation. Select the first document by double-clicking on the link.

3. Login with your ¡username¿ and ¡password¿. The login dialog can be opened by moving your mouse over the top bar of the interface and then by clicking 'login' (top right). Your usernames and passwords are:

B. Subsequently, the annotation routine is as follows:

1. The string (i.e. the text segment), which marks the content of a frame needs to be highlighted. These strings can span over one or several words and may include fragments (several non-successive words).

2. The highlighted text fragments need to be labeled with one of the categories for both substance and evaluation as outlined above.

3. Finally, if possible, draw an arrow a frame to an entity.

4. Don't forget to include comments or questions in the 'Notes' textbox if you have any.

5. When you are done which all annotations in a document, you can select the next one by clicking on the next arrow in the top bar of the interface.

PLEASE NOTE: If you feel insecure when making a decision on either the substance or the evaluation of a frame, please indicate this as follows:

1. Choose the label for substance or evaluation you believe to be the appropriate one, despite your insecurity.

2. Add a comment in the following format: "NOT SURE: substance, Transparency OR Participation; evaluation, normative$_{l}$ess OR factual$_{m}$issing". Please use exactly this template, especially the 'NOT SURE' and 'OR', the colons, semicolons and commas. Also, please use the labels exactly as indicated in the interface, e.g. "Participation" not "participation" or "participating". If your insecurity only refers to one aspect of a frame, leave the other aside.

PLEASE NOTE: It is not problem to annotate overlapping frames, i.e. the same text passage can be annotated multiple times if you decide that it refers to multiple frames.

[end of codebook]

C.2 Intercoder Reliability

After an intensive training phase, the measured inter-annotator agreement was constantly high (micro-averaged pair-wise F1-scores for fine-grained frame categories ranged between 0.66 for 23 documents during the pre-final test phase and 0.71 for 5 documents at the start of the annotation).

Since the second evaluation set was only coded by a single annotator, we cannot report any reliability measurements here.

Bibliography

Adi, Y., Kermany, E., Belinkov, Y., Lavi, O., and Goldberg, Y. (2016). Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*.

Adi, Y., Kermany, E., Belinkov, Y., Lavi, O., and Goldberg, Y. (2017). Analysis of sentence embedding models using prediction tasks in natural language processing. *IBM Journal of Research and Development*, 61(4/5):3:1–3:9.

Aggarwal, C. C. and Zhai, C. (2012). A survey of text classification algorithms. In *Mining text data*, pages 163–222. Springer.

Altınel, B. and Ganiz, M. C. (2018). Semantic text classification: A survey of past and recent advances. *Information Processing & Management*, 54(6):1129–1153.

Amsler, M., Wüest, B., and Schneider, G. (2016). Legitimacy of new forms of governance in public discourse - an automated media content analysis approach driven by techniques of computational linguistics. In *Proceedings of PolText 2016. International Conference on the Advances in Computational Analysis of Political Text*, pages 1–7.

An, J., Kwak, H., and Ahn, Y.-Y. (2018). Semaxis: A lightweight framework to characterize domain-specific word semantics beyond sentiment. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2450–2461.

Arora, S., Li, Y., Liang, Y., Ma, T., and Risteski, A. (2016). A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Arora, S., Li, Y., Liang, Y., Ma, T., and Risteski, A. (2018). Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495.

Arora, S., Liang, Y., and Ma, T. (2017). A simple but tough-to-beat baseline for sentence embeddings. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Artetxe, M., Labaka, G., and Agirre, E. (2017). Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462.

Artetxe, M., Labaka, G., and Agirre, E. (2018). A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798.

Astrup, C. (1978). Physiological mechanisms of flooding (implosion) therapy. *The Pavlovian Journal of Biological Science: Official Journal of the Pavlovian*, 13(4):195–198.

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 2200–2204.

Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 238–247.

Belinkov, Y. and Glass, J. (2019). Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Berger, P. L. and Luckmann, T. (1991). *The Social Construction of Reality: A Treatise in the Sociology of Knowledge*. Penguin UK.

Bholat, D., Hansen, S., Santos, P., and Schonhardt-Bailey, C. (2015). Text mining for central banks. Technical report, Centre for Central Banking Studies, Bank of England.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, pages 4349–4357.

Brody, S. and Diakopoulos, N. (2011). Cooooooooooooooollllllllllllll!!!!!!!!!!!!!!: using word lengthening to detect sentiment in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 562–570. Association for Computational Linguistics.

Caliskan, A., Bryson, J. J., and Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Camacho-Collados, J. and Pilehvar, M. T. (2018). From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*, 63:743–788.

Cheng, A.-S., Fleischmann, K. R., Wang, P., and Oard, D. W. (2008). Advancing social science research by applying computational linguistics. *Proceedings of the American Society for Information Science and Technology*, 45(1):1–12.

Chong, D. and Druckman, J. N. (2007a). Framing theory. *Annual Review of Political Science*, 10(1):103–126.

Chong, D. and Druckman, J. N. (2007b). A theory of framing and opinion formation in competitive elite environments. *Journal of Communication*, 57(1):99–118.

Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Clematide, S. and Klenner, M. (2010). Evaluation and extension of a polarity lexicon for german. In *Proceedings of the First Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 7–13.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine learning*, pages 160–167. ACM.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., and Baroni, M. (2018). What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136.

D'Angelo, P. and Kuypers, J. A. (2010). *Doing News Framing Analysis: Empirical and Theoretical Perspectives*. Routledge.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dhillon, I. S. and Modha, D. S. (2001). Concept decompositions for large sparse text data using clustering. *Machine learning*, 42(1-2):143–175.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Eisenstein, J. (2013). What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369.

Entman, R. M. (1993). Framing: Toward clarification of a fractured paradigm. *Journal of communication*, 43(4):51–58.

Erk, K. (2012). Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653.

Esuli, A. and Sebastiani, F. (2006). SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, pages 417–422.

Ethayarajh, K., Duvenaud, D., and Hirst, G. (2019). Towards understanding linear word analogies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3253–3262, Florence, Italy. Association for Computational Linguistics.

Ettinger, A., Elgohary, A., Phillips, C., and Resnik, P. (2018). Assessing composition in sentence vector representations. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1790–1801.

Eugenio, B. D. and Glass, M. (2004). The kappa statistic: A second look. *Computational Linguistics*, 30(1):95–101.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2015). Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615.

Faruqui, M., Tsvetkov, Y., Rastogi, P., and Dyer, C. (2016). Problems with evaluation of word embeddings using word similarity tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 30–35. Association for Computational Linguistics.

Feinstein, A. R. and Cicchetti, D. V. (1990). High agreement but low kappa: I. the problems of two paradoxes. *Journal of Clinical Epidemiology*, 43(6):543–549.

Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. In *Studies in Linguistic Analysis (Special Volume of the Philological Society)*, pages 1–32. Basil Blackwell, Oxford.

Gittens, A., Achlioptas, D., and Mahoney, M. W. (2017). Skip-gram- zipf+ uniform= vector additivity. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 69–76.

Goffman, E. (1974). *Frame Analysis: An essay on the Organization of Experience.* Harvard University Press.

Goldberg, Y. (2017). Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1):1–309.

Goldberg, Y. and Levy, O. (2014). word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Grimmer, J. and Stewart, B. M. (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political analysis*, 21(3):267–297.

Gwet, K. L. (2014). *Handbook of Inter-rater Reliability: The Definitive Guide to Measuring the Extent of Agreement Among Multiple Raters.* Advanced Analytics, LLC.

Gyllensten, A. C., Ekgren, A., and Sahlgren, M. (2019). R-grams: Unsupervised learning of semantic units in natural language. In *Proceedings of the 13th International Conference on Computational Semantics - Student Papers*, pages 52–62, Gothenburg, Sweden. Association for Computational Linguistics.

Gyllensten, A. C. and Sahlgren, M. (2015). Navigating the semantic horizon using relative neighborhood graphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2451–2460.

Gyllensten, A. C. and Sahlgren, M. (2018). Distributional term set expansion. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., and Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239.

Hamilton, W. L., Clark, K., Leskovec, J., and Jurafsky, D. (2016). Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 595–605. NIH Public Access.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

Hatzivassiloglou, V. and McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In *Proc. of the ACL 1997*, pages 174–181. Association for Computational Linguistics.

Hilbert, M., Barnett, G., Blumenstock, J., Contractor, N., Diesner, J., Frey, S., González-Bailón, S., Lamberson, P., Pan, J., Peng, T.-Q., et al. (2019). Computational communication science— computational communication science: A methodological catalyzer for a maturing discipline. *International Journal of Communication*, 13:23.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177. ACM.

Huang, E., Socher, R., Manning, C., and Ng, A. (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–882. Association for Computational Linguistics.

Huang, S., Niu, Z., and Shi, C. (2014). Automatic construction of domain-specific sentiment lexicon based on constrained label propagation. *Knowledge-Based Systems*, 56:191–200.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning*, pages 137–142. Springer.

Joulin, A., Bojanowski, P., Mikolov, T., Jégou, H., and Grave, E. (2018). Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T. (2016). Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431.

Jurafsky, D. and Martin, J. H. (2019). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (Third Edition draft)*. Online updated version of the book, retrieved from `https://web.stanford.edu/~jurafsky/slp3/` (Nov. 2019).

Kanayama, H. and Nasukawa, T. (2006). Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 355–363. Association for Computational Linguistics.

Karlsson, F. (1990). Constraint grammar as a framework for parsing running text. In *COLNG 1990 Volume 3: Papers presented to the 13th International Conference on Computational Linguistics*.

Karlsson, F., Anttila, A., Voutilainen, A., and Heikkilae, J. (1995). *Constraint Grammar: A Language-independent System for Parsing Unrestricted Text*. Walter de Gruyter.

Khodak, M., Saunshi, N., Liang, Y., Ma, T., Stewart, B., and Arora, S. (2018). A la carte embedding: Cheap but effective induction of semantic feature vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22.

Kiela, D., Hill, F., and Clark, S. (2015). Specializing word embeddings for similarity or relatedness. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2044–2048.

Kim, H. K., Kim, H., and Cho, S. (2017). Bag-of-concepts: Comprehending document representation through clustering words in distributed representation. *Neurocomputing*, 266:336–352.

Kim, S.-M. and Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*, page 1367. Association for Computational Linguistics.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3294–3302.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966.

Lample, G., Conneau, A., Ranzato, M., Denoyer, L., and Jégou, H. (2018). Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Lau, J. H. and Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 78–86. Association for Computational Linguistics.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Lenci, A. (2018). Distributional models of word meaning. *Annual Review of Linguistics*, 4:151–171.

Levy, O. and Goldberg, Y. (2014a). Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 171–180. Association for Computational Linguistics.

Levy, O. and Goldberg, Y. (2014b). Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185.

Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Lipton, Z. C. (2018). The mythos of model interpretability. *Queue*, 16(3):31–57.

Liu, B. and Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In *Mining Text Data*, pages 415–463. Springer.

Lowe, W. (2001). Towards a theory of semantic space. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 23.

Luong, T., Socher, R., and Manning, C. (2013). Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113.

Manning, C., Raghavan, P., and Schütze, H. (2010). Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103.

Manning, C. D. (2015). Computational linguistics and deep learning. *Computational Linguistics*, 41(4):701–707.

Marcus, G. (2018). Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*.

Matthes, J. (2014a). *Framing*. Nomos.

Matthes, J. (2014b). Zum Gehalt der Framing-Forschung: Eine kritische Bestandsaufnahme. In *Framing als politischer Prozess*, pages 15–29. Nomos.

McCallum, A., Nigam, K., et al. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In Bengio, Y. and LeCun, Y., editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., and Joulin, A. (2018). Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Mirończuk, M. M. and Protasiewicz, J. (2018). A recent overview of the state-of-the-art elements of text classification. *Expert Systems with Applications*, 106:36–54.

Neuendorf, K. A. (2016). *The Content Analysis Guidebook*. Sage.

Neviarouskaya, A., Prendinger, H., and Ishizuka, M. (2011). Sentiful: A lexicon for sentiment analysis. *Affective Computing, IEEE Transactions on*, 2(1):22–36.

Osgood, C. E., Suci, G. J., and Tannenbaum, P. H. (1957). *The Measurement of Meaning*. University of Illinois press.

Pagliardini, M., Gupta, P., and Jaggi, M. (2018). Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. In *NAACL 2018 - Conference of the North American Chapter of the Association for Computational Linguistics*.

Patel, A., Sands, A., Callison-Burch, C., and Apidianaki, M. (2018). Magnitude: A fast, efficient universal vector embedding utility package. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 120–126.

Pennebaker, J. W., Boyd, R. L., Jordan, K., and Blackburn, K. (2015). The development and psychometric properties of LIWC2015. Technical report, The University of Texas at Austin.

Pennebaker, J. W., Francis, M. E., and Booth, R. J. (2001). Linguistic inquiry and word count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Petchler, R. and González-Bailón, S. (2015). Automated content analysis of online political communication. In Coleman, S. and Freelon, D., editors, *Handbook of Digital Politics*. Edward Elgar Publishing.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Qiu, G., Liu, B., Bu, J., and Chen, C. (2011). Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37(1):9–27.

Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50.

Reisinger, J. and Mooney, R. J. (2010). Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117. Association for Computational Linguistics.

Riloff, E. and Wiebe, J. (2003). Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 105–112.

Rocchio, J. (1971). Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing*, pages 313–323.

Rothe, S., Ebert, S., and Schütze, H. (2016). Ultradense word embeddings by orthogonal transformation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 767–777.

Ruder, S. (2019). *Neural Transfer Learning for Natural Language Processing*. PhD thesis, National University of Ireland, Galway.

Sahlgren, M. (2006). *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. PhD thesis, Stockholm University.

Sahlgren, M. and Cöster, R. (2004). Using bag-of-concepts to improve the performance of support vector machines in text categorization. In *Proceedings of the 20th International Conference on Computational Linguistics*, page 487. Association for Computational Linguistics.

Sahlgren, M. and Lenci, A. (2016). The effects of data size and frequency range on distributional semantic models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 975–980. Association for Computational Linguistics.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

San Vicente, I., Agerri, R., and Rigau, G. (2014). Simple, robust and (almost) unsupervised generation of polarity lexicons for multiple languages. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 88–97.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK.

Schmidt, V. A. (2013). Democracy and legitimacy in the european union revisited: Input, output and 'throughput'. *Political Studies*, 61(1):2–22.

Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Schwartz, H. A. and Ungar, L. H. (2015). Data-driven content analysis of social media: a systematic overview of automated methods. *The ANNALS of the American Academy of Political and Social Science*, 659(1):78–94.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Severyn, A. and Moschitti, A. (2015). On the automatic learning of sentiment lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1397–1402.

Shah, D. V., Cappella, J. N., and Neuman, W. R. (2015). Big data, digital media, and computational social science: Possibilities and perils. *The ANNALS of the American Academy of Political and Social Science*, 659(1):6–13.

Shen, D., Wang, G., Wang, W., Min, M. R., Su, Q., Zhang, Y., Li, C., Henao, R., and Carin, L. (2018). Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–450.

Speer, R., Chin, J., and Havasi, C. (2017). ConceptNet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4444–4451. AAAI Press.

Speer, R., Chin, J., Lin, A., Jewett, S., and Nathan, L. (2018). Luminosoinsight/wordfreq: v2.2.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J. (2012). brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107. Association for Computational Linguistics.

Stone, P. J., Dunphy, D. C., Smith, M. S., and Ogilvie, D. M. (1966). *The General Inquirer: A Computer Approach to Content Analysis.* MIT press.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307.

Tang, D., Wei, F., Qin, B., Zhou, M., and Liu, T. (2014). Building large-scale twitter-specific sentiment lexicon: A representation learning approach. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 172–182.

Teufel, S. (1999). *Argumentative zoning: Information extraction from scientific text.* PhD thesis, University of Edinburgh.

Trask, A., Michalak, P., and Liu, J. (2015). sense2vec - a fast and accurate method for word sense disambiguation in neural word embeddings. *arXiv preprint arXiv:1511.06388*.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL '02)*, pages 417–424. Association for Computational Linguistics.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Tversky, A. (1977). Features of similarity. *Psychological Review*, 84(4):327.

Van Atteveldt, W. and Peng, T.-Q. (2018). When communication meets computation: Opportunities, challenges, and pitfalls in computational communication science. *Communication Methods and Measures*, 12(2-3):81–92.

Van Atteveldt, W., Strycharz, J., Trilling, D., and Welbers, K. (2019). Computational communication science— toward open computational communication science: A practical road map for reusable data and code. *International Journal of Communication*, 13:20.

Velikovich, L., Blair-Goldensohn, S., Hannan, K., and McDonald, R. (2010). The viability of web-derived polarity lexicons. In *Human Language Technologies: The 2010*

Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 777–785. Association for Computational Linguistics.

Vicente, I. S. and Saralegi, X. (2016). Polarity lexicon building: to what extent is the manual effort worth? In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 938–942.

Wang, L. and Xia, R. (2017). Sentiment lexicon construction with representation learning based on hierarchical sentiment supervision. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 502–510.

Wang, S. and Manning, C. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–94. Association for Computational Linguistics.

Wieting, J., Bansal, M., Gimpel, K., and Livescu, K. (2016a). Charagram: Embedding words and sentences via character n-grams. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1515, Austin, Texas. Association for Computational Linguistics.

Wieting, J., Bansal, M., Gimpel, K., and Livescu, K. (2016b). Towards universal paraphrastic sentence embeddings. In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

Wongpakaran, N., Wongpakaran, T., Wedding, D., and Gwet, K. L. (2013). A comparison of Cohen's Kappa and Gwet's AC1 when calculating inter-rater reliability coefficients: a study conducted with personality disorder samples. *BMC Medical Research Methodology*, 13(1):61.

Wüest, B., Amsler, M., and Schneider, G. (2017). SIFT—a language technology toolkit to assess the print media coverage of new forms of governance. Working Paper 95, University of Zurich. NCCR-Democracy.

Zamith, R. and Lewis, S. C. (2015). Content analysis and the algorithmic coder: What computational social science means for traditional modes of media analysis. *The ANNALS of the American Academy of Political and Social Science*, 659(1):307–318.

Zesch, T. and Gurevych, I. (2006). Automatically creating datasets for measures of semantic relatedness. In *Proceedings of the Workshop on Linguistic Distances*, pages 16–24. Association for Computational Linguistics.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, pages 649–657.

Zipf, G. K. (1949). *Human Behavior and the Principle of Least Effort.* Addison-Wesley Press, Cambridge, (Mass.).